Proyecto Fin de Máster

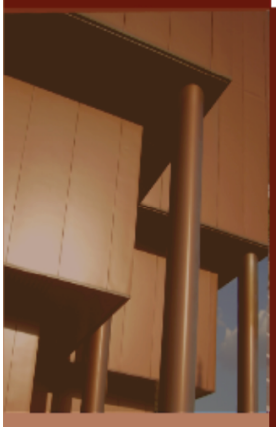
Máster en Sistemas de Energía Eléctrica

Análisis de los Métodos de Predicción Aplicados a los Desvíos en el Sistema Eléctrico Ibérico

Autor: Carlos Rodríguez Manzanares

Tutores: Jesús Manuel Riquelme Santos

Juan Manuel Roldán Fernández



Departamento de Ingeniería Eléctrica

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2018

Proyecto Fin de Máster
Máster en Sistemas de Energía Eléctrica

Análisis de los Métodos de Predicción Aplicados a los Desvíos en el Sistema Eléctrico Ibérico

Autor:

Carlos Rodríguez Manzanares

Tutores:

Jesús Manuel Riquelme Santos

Catedrático de Universidad

Juan Manuel Roldán Fernández

Doctor Ingeniero Industrial

Departamento de Ingeniería Eléctrica

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2018

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

RESUMEN

Actualmente, la estructuración de la generación-demanda en el Sistema Eléctrico Español se organiza mediante mecanismos de mercado. En estos mercados, acuden tanto consumidores como productores con el fin de realizar sus ofertas de compraventa de energía, creándose finalmente un compromiso de todos ellos de satisfacer los tratos realizados consumiendo o generando la energía pactada.

Esto no sería posible de llevar a cabo sin un respaldo del operador del sistema, gestionando mecanismos de reserva y regulación para corregir los déficits o excesos que se creen derivados de la incertidumbre que existe al tratar con energías renovables como la fotovoltaica, la eólica o el comportamiento, no siempre predecible, de la demanda. Todo sujeto de mercado que no pueda asegurar satisfacer el acuerdo en su totalidad es candidato a recibir penalizaciones económicas.

La realización del presente proyecto surge como continuación de un trabajo inicial titulado Análisis de los Desvíos en el Sistema Eléctrico Ibérico y sus Implicaciones Económicas y resultará útil tener cierto conocimiento de las conclusiones que en él se obtienen para un mejor seguimiento de este estudio. En dicho trabajo se utilizan datos reales de distintos parámetros del Sistema Eléctrico Español para llevar a cabo un estudio del volumen de energía desviada junto con sus pérdidas económicas asociadas. Se analiza qué sujetos de mercado intervienen en el problema y en qué medida lo hacen, se describen patrones y tendencias y se desarrollan estrategias con el fin de reducir dichos desvíos.

Más allá del desarrollo de estrategias que ayuden a los ofertantes a ver disminuidas sus penalizaciones, lo que busca este estudio es el desarrollo de metodologías que ayuden a predecir el sentido de desvío del sistema con un día de antelación. Esto aportará información muy valiosa a aquellos sujetos de mercado afectados, posibilitándoles el modificar sus ofertas de la manera oportuna y hacer una mejor gestión de la energía producida o consumida.

1. INTRODUCCIÓN

Para comprender el motivo por el que se producen los desvíos generación-demanda y poder evaluar los efectos que estos tienen en el sistema eléctrico, es necesario conocer en cierta medida la estructura del mercado de producción y comprender cómo se organizan cada una de sus secciones y de qué manera se realiza la liquidación económica de cada uno de los sujetos que a él acuden.

De esto se hizo una descripción completa en el anterior trabajo y no resulta de interés el volver a hacerlo. Sólo se presentará parte de los resultados finales a modo introductorio, justificando el actual estudio y se recordará, a modo de resumen, información sobre los mercados que sea especialmente importante para el mismo.

1.1. RESULTADOS DEL ANÁLISIS DE LOS DESVÍOS

El análisis del trabajo inicial abarca el estudio de los años 2014, 2015 y 2016. En la siguiente tabla (*tabla 1.1*) se encuentran datos reales acerca del volumen de energía desviada y sus costes asociados durante estos tres períodos.

Variable\Período	2014	2015	2016
Energía desviada a bajar (MWh)	5.076.961	5.196.230	3.674.288
Costes totales desvíos a bajar (€)	62.058.027	61.719.444	31.597.263
Nº de horas a bajar	5883	5600	5360
Potencia media a bajar cada hora (MW)	863	928	685
Energía desviada a subir (MWh)	1.784.215	2.042.644	1.777.038
Costes totales desvíos a Subir (€)	45.918.334	46.500.970	28.050.055
Nº de horas a subir	2876	3160	3400
Potencia media a subir cada hora (MW)	620	646	523

Tabla 1.1 - Energía desviada y sobrecostes globales del sistema

Como se puede observar, las pérdidas económicas a nivel global cada año, ascienden al orden de millones de euros y los valores medios de potencia desviada a cada hora pueden aproximarse a los valores de potencia instalada de una central convencional de dimensiones importantes. Esto último a efectos prácticos, se puede entender como qué el sistema debe ser capaz de absorber la conexión o desconexión de una central de estas características partiendo de un estado inicial de equilibrio. Por supuesto, se trata de una media y se entiende que tanto habrá horas en las que los desvíos globales sean menores, como por el contrario, habrá otras en las que sean mucho mayores. Con todo esto, lo que se pretende dejar claro es, que conseguir una mejora en la gestión de los desvíos resulta tan importante y necesario, tanto a nivel económico como a nivel de seguridad del sistema.

En la tabla siguiente (*tabla 1.2*) se pueden ver las pérdidas económicas asociadas a la demanda, a la producción eólica y a la producción fotovoltaica, siendo estos principalmente los participantes del mercado que se encargarán de cometer la mayor parte de los desvíos debido a la naturaleza propia de cada uno.

Variable\Período	2014	2015	2016
Costes desvíos a bajar de la demanda(€)	32.075.156	31.809.034	11.915.359
Costes desvíos a subir de la demanda(€)	18.357.505	21.735.350	10.751.391
Costes desvíos a bajar de la eólica (€)	15.129.663	17.953.101	12.364.606
Costes desvíos a subir de la eólica(€)	16.966.292	14.496.341	8.765.935
Costes desvíos a bajar de la fotovoltaica (€)	4.896.855	2.886.026	2.276.493
Costes desvíos a subir de la fotovoltaica (€)	3.866.018	3.716.790	2.734.140

Tabla 1.2 - Coste de los desvíos de la demanda, eólicos y fotovoltaicos.

Con esta información, se deja claro finalmente, quienes serían los principales interesados en contar con nuevas herramientas predictivas que les ayudasen en la toma de decisiones a la hora de realizar sus ofertas en el mercado.

1.2. MERCADO DE PRODUCCIÓN

Del mercado de producción, se conoce que es gestionado por dos organismos. El primero será el Operador del Mercado, (OMIE para el caso español), al cual le corresponde controlar el Mercado Diario y las seis sesiones de Mercado Intradiario. El segundo organismo es el Operador del Sistema (Red Eléctrica de España), al cual le corresponde la responsabilidad de gestionar los Servicios de Ajuste y Balance, dentro de los cuales encontramos: la Regulación Secundaria, la

Regulación Terciaria, la Resolución de las Restricciones Técnicas, el establecimiento de la Reserva de Potencia a Subir y la Gestión de Desvíos.

Toda la descripción funcional de cada una de las partes que componen el mercado se realizó en el anterior trabajo, por lo tanto, lo que realmente interesa ahora, es recordar en qué momento y en qué orden se suceden cada uno de los eventos que componen este mercado. Todo ello, con la finalidad de saber con qué información es posible contar en cada instante, con el objetivo de poder utilizarla cuando aún sea posible modificar las ofertas realizadas para el día siguiente.

Si se observa la siguiente tabla (*tabla 1.3*), se pueden ver los distintos eventos que componen cada sesión de Mercado Intradiario. La primera sesión, realiza su apertura a las 17:00 horas y recoge ofertas hasta las 18:45 horas. A las 19:30 horas se lleva a cabo la casación y se resuelven las restricciones técnicas. A las 20:20 horas se publica el PHF (Programa Horario Final), que contiene los resultados de todos los mercados anteriores hasta ese momento. El horizonte de programación de esta primera sesión de Mercado Intradiario abarca desde las 21:00 horas del día en curso hasta las 23:00 horas del día siguiente, con lo que resulta un buen recurso para modificar las ofertas realizadas en el Mercado Diario, ya que es posible re-ofertar para la totalidad de las horas del día siguiente. En la segunda sesión de Mercado Intradiario, será todavía posible ofertar para la totalidad de las horas del día siguiente y en general, el resto de posteriores sesiones, también podrán ser utilizadas como un recurso más para modificar posteriores ofertas dentro de sus respectivos horizontes de programación. Aun así, la estrategia que se planteará en este trabajo será la de realizar una predicción con la información que se tenga hasta justo antes de la apertura de la primera sesión, con el objetivo de que sea a partir de esta, cuando sea posible contar con información fiable acerca de los sentidos de los desvíos del día siguiente.

	Sesión1	Sesión2	Sesión3	Sesión4	Sesión5	Sesión6
Apertura de sesión	17:00	21:00	1:00	4:00	8:00	12:00
Cierre de sesión	18:45	21:45	1:45	4:45	8:45	12:45
Casación	19:30	22:30	2:30	5:30	9:30	13:30
Restricciones técnicas	20:10	23:10	3:10	6:10	10:10	14:10
Publicación PHF	20:20	23:20	3:20	6:20	10:20	14:20
Horizonte de programación	27 horas	24 horas	20 horas	17 horas	13 horas	9 horas
Períodos horarios	22*-24	1-24	5-24	8-24	12-24	16-24

*Referida a la hora 22 del día D-1

Tabla 1.3 - Sesiones de Mercado Intradiario

Antes de las 17:00 horas, ya será posible contar con una cantidad importante de información como para poder realizar una aproximación de lo que ocurrirá al día siguiente. El PDVP (Programa Diario Viable Provisional), es publicado a las 16:00 horas y se compone del resultado completo del Mercado Diario y de la Resolución de Restricciones Técnicas. Paralelamente a la publicación del PDVP, también se fijan, por parte del Operador del Sistema, los requerimientos de Reserva de Potencia a Subir y de Regulación Secundaria para cada una de las horas del día

siguiente. Por tanto, esta será la información relativa al sistema eléctrico con la que se cuente antes de la primera sesión de intradiario.

Por otro lado, no solo resulta importante contar con información que sea relativa a variables del sistema exclusivamente. Es bien conocido que las condiciones meteorológicas también tienen un efecto importante sobre la demanda y la producción eléctrica, por lo que será interesante contar con las previsiones meteorológicas que se hayan realizado hasta ese momento para el día siguiente.

2. METODOLOGÍA

Tal como se ha adelantado en la introducción, se pretende predecir lo que ocurrirá en las 24 horas del día próximo con la información que se tenga antes de la apertura de la primera sesión de Mercado Intradiario.

Sujetos de mercado tales como centrales eólicas, fotovoltaicas o los consumidores, saben que sus previsiones son aproximadas y que siempre cometerán errores en la programación por pequeños que sean. Si se piensa desde el punto de vista de uno de estos sujetos, el cual cuenta con una previsión de consumo o generación junto con un umbral de incertidumbre para las 24 horas del día siguiente, resultaría interesante conocer el sentido del desvío del sistema para cada una de esas horas. Ello permitiría introducir el margen de incertidumbre en la oferta buscando el desvío propio hacia el sentido que sea favorable al sistema. De tal manera que si se predice que el sistema se desviará a bajar, el sujeto se desvíe a subir y viceversa. En caso de acierto en la predicción, una central generadora debería entrar en uno de los dos casos siguientes:

- El sistema se ha desviado a bajar y la central ha conseguido desviarse a subir: se le paga a la central por la energía generada de más, tal como si la hubiera introducido en mercado.

- El sistema se ha desviado a subir y la central ha conseguido desviarse a bajar: la central debe devolver la energía vendida en mercado al mismo precio que la vendió.

Por lo tanto, partiendo de la información previa a la primera sesión de Mercado Intradiario, el objetivo será el de crear un método de predicción que genere un vector de 24 valores, pudiendo ser estos: 1 ó -1. Estos valores serán la representación del sentido del desvío del sistema para cada una de las 24 horas del día siguiente.

2.1. WEKA

WEKA es el nombre que recibe el software que se utilizará para la realización del estudio, cuyo acrónimo recibe el nombre de *Waikato Environment for Knowledge Analysis.* Se trata de un programa de libre distribución desarrollado por la universidad de Waikato, de Nueva Zelanda. Está escrito en JAVA y se utiliza generalmente para tareas de aprendizaje automático y minería de datos.

Es cierto que el programa tiene cierta complejidad y gran cantidad de funciones, pero aquí se presentará solo la parte que guarde estricta relación con la serie de tareas que se quieren llevar a cabo.

2.1.1. ENTRADA AL PROGRAMA

Tras abrir el programa, se abrirá una ventana en la cual encontraremos una serie de pestañas desplegables en la esquina superior izquierda y una columna en el margen izquierdo con 5 sub-herramientas distintas. Dentro del alcance del siguiente estudio solo será necesario aprender la utilización del *Explorer* y del *Experimenter*.

Figura 2.1 - Ventana inicial de WEKA

2.1.2. EL EXPLORER

El *Explorer* permite visualizar y aplicar distintos algoritmos de aprendizaje a un conjunto de datos. Concretamente servirá para aplicar estos algoritmos a datos reales para tratar de predecir el sentido del desvío del sistema. Cada una de las tareas de la minería de datos que es posible aplicar viene representada por una pestaña en la parte superior. Estas son:

- *Preprocess*: permite la visualización y el pre-procesado de la información que será objeto de estudio. Se podrán aplicar filtros a estos datos y escoger que variables formarán o no parte del estudio. Por supuesto, esta información ha de ser introducida a WEKA en un formato especial que se explicará más adelante.

- *Classify:* este apartado resultará uno de los más importantes. Aquí es donde se encuentran los distintos algoritmos de clasificación y regresión con los que cuenta el programa. Será posible escoger el método de predicción que se desee, modificar sus parámetros, cambiar el modo de entrenamiento y testeo, realizar pruebas con el conjunto de datos que se esté estudiando, visualizar resultados y exportar modelos creados y volverlos a importar para aplicarlos a otros conjuntos de datos.

- *Cluster y Associate:* estas dos funcionalidades no se han llegado a utilizar en el estudio pero sirven para crear agrupaciones y asociaciones de los datos de entrada al programa.

- *Select Attributes:* esta también será una herramienta de cierta importancia, ya que permitirá realizar clasificaciones y selección de los datos proporcionados como entrada al programa. Para ello, utilizará los mismos algoritmos de aprendizaje y clasificación que se pueden encontrar en *Classify*. Esto ayuda a saber qué atributos, de los que se han dado como entrada, son importantes para realizar la predicción deseada. Ayuda a acotar el estudio, a eliminar variables que puedan falsear el resultado y a reducir los tiempos de computación.

- *Visualize:* en este apartado es posible visualizar la información dada como entrada al programa por parejas de atributos.

- *Experimenter*: esta herramienta no forma parte del *Explorer*, pero se incluirá en este apartado simplemente para que sea nombrada, ya que será explicada más exhaustivamente al final de la metodología. Con ella básicamente se podrán programar experimentos más complejos utilizando los algoritmos y funciones aprendidas en el *Explorer*.

Hasta aquí, se ha hecho una descripción general de las principales funciones que se utilizarán en el estudio. A continuación se presentará la información que se ha decidido utilizar, de dónde se ha conseguido, como ha sido tratada para ser introducida en WEKA y la realización de pruebas con estos datos a modo de ejemplo que sirva para mostrar, de manera más profunda, las posibilidades del programa.

2.2. DATOS DE PARTIDA

Como ya se ha adelantado en la introducción, como datos de entrada se utilizarán: los resultados publicados del PDVP para cada tecnología, los requerimientos de reserva de potencia a subir, los requerimientos de regulación secundaria e información de ciertas variables meteorológicas de lugares concretos de la Península. Las páginas de referencia para la descarga de datos serán www.esios.ree.es en el caso de información relacionada con el sistema eléctrico, y www.meteomanz.com para la meteorológica.

A continuación, se enumeran los distintos ficheros que se han descargado de www.esios.ree.es, con información horaria de distintas variables desde el año 2014 al 2017, ambos inclusive:

- Resultado del PDVP para la demanda.
- Resultado del PDVP para el enlace de Baleares.
- Resultado del PDVP para las interconexiones internacionales.
- Resultado del PDVP para bombeo en centrales hidráulicas de bombeo.
- Resultado del PDVP para turbinación en centrales hidráulicas de bombeo.
- Resultado del PDVP para centrales de carbón.
- Resultado del PDVP para centrales de ciclo combinado.
- Resultado del PDVP para centrales de cogeneración.
- Resultado del PDVP para las centrales nucleares.
- Resultado del PDVP para centrales hidráulicas.
- Resultado del PDVP para centrales eólica.
- Resultado del PDVP para centrales fotovoltaicas.
- Resultado del PDVP para centrales termosolares.
- Precio del mercado diario.
- Requerimientos de reserva de potencia a subir.
- Requerimientos de regulación secundaria a bajar.
- Requerimientos de regulación secundaria a subir.
- Sentido del desvío del sistema.
- Sentido del desvío de la demanda.
- Sentido del desvío eólico.
- Sentido del desvío fotovoltaico.
- Sentido de la gestión de desvíos.

A continuación, se enumeran los distintos ficheros que se han descargado de www.meteomanz.com, con información horaria de distintas variables meteorológicas desde el año 2014 al 2017, ambos inclusive:

- Temperatura en Madrid.
- Velocidad del viento en Madrid.
- Humedad relativa en Madrid.
- Presión atmosférica en Madrid.
- Temperatura en Vigo.
- Velocidad del viento en Vigo.
- Humedad relativa en Vigo.
- Presión atmosférica en Vigo.
- Temperatura en Gibraltar.
- Velocidad del viento en Gibraltar.
- Humedad relativa en Gibraltar.
- Presión atmosférica en Gibraltar.

Son estas cuatro variables meteorológicas las que se han elegido porque son las que presentan un formato numérico más fácil de tratar o porque básicamente, se proporcionan valores de ellas para casi la totalidad de las horas de los 4 años que componen el estudio. En la página también se pueden encontrar otras variables como la nubosidad o el sentido del viento, pero se han desechado, ya sea por entender que no proporcionan información adicional o porque falten gran cantidad de datos.

Se puede encontrar esta información para estaciones meteorológicas repartidas por todo el mundo. En este caso, tratándose de España, se han escogido Madrid, Vigo y Gibraltar por varios motivos. Por un lado, se sabe que la demanda es la que más desvíos comete, seguida de la generación eólica. También se conoce que tanto el consumo eléctrico como la generación renovable dependen en gran medida de factores meteorológicos como la temperatura o el viento. Por lo tanto se han buscado lugares donde la demanda sea importante, como Madrid, y lugares con alta densidad de centrales eólicas, como son Galicia y la provincia de Cádiz. Así será posible encontrar una relación lógica o un mayor impacto entre estas variables meteorológicas y las variables del sistema eléctrico.

En el anterior proyecto ya se explicaba como descargar la información de la página de REE. Así que se explicará simplemente como funciona la web de *Meteomanz*. A continuación se puede ver (*figura 2.2*) el aspecto que tiene el inicio de la web. Se ha remarcado en rojo la zona que resulta de interés.

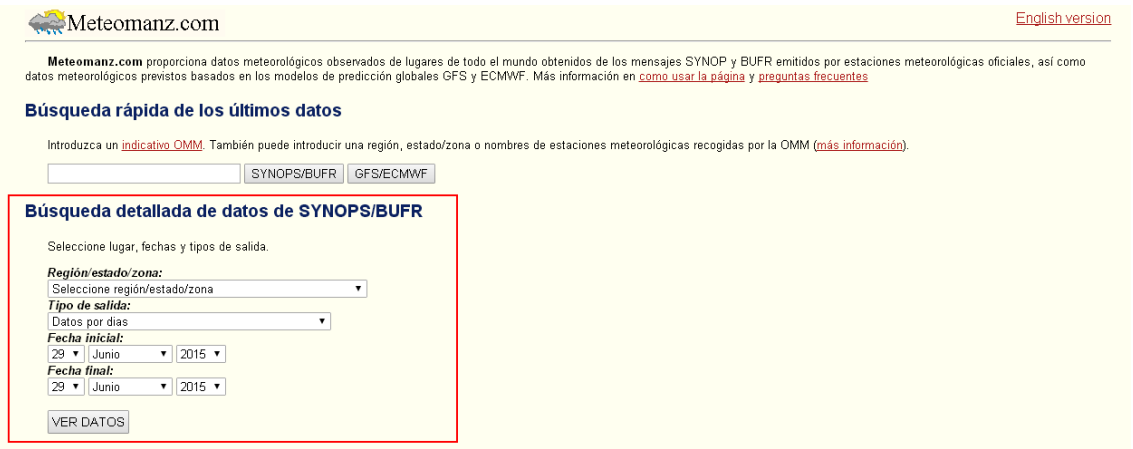


Figura 2.2 - Inicio de la web de www.meteomanz.com

Se ha de seleccionar la región, estado o zona de la que se quiere extraer la información. En este caso se seleccionará *España*. En la pestaña *Tipo de salida*, se escogerá la opción *Mapas* y en *Tipo de mapa* se seleccionará la opción *Mapa de nombre de estaciones*. Esto nos permite ver la distribución de estaciones meteorológicas que se encuentran disponibles en la web, junto con su nombre, coordenadas y otra serie de datos que se pueden visualizar accediendo a cada una de ellas.

Figura 2.3 - Distribución de estaciones españolas disponibles en Meteomanz

Una vez visto como están distribuidas las estaciones y ver cuáles están disponibles, se vuelve a la ventana inicial y se vuelve a seleccionar *España*, pero esta vez se escoge la estación que se desee en la pestaña *Estación*. Se seleccionará *Datos por horas (variables principales)* en la pestaña de *Tipo de salida* y se elegirá fecha y hora de inicio y final. La página solo permite descargar de una sola vez la información relativa a un mes, con lo que habrá que hacerlo mes a mes hasta ir completando la información para los cuatro años de estudio. Es posible elegir la descarga en formato Excel para poder tratar los ficheros más fácilmente con Matlab. Resulta también recomendable, crear una función que lea la información de los distintos meses y devuelva a la salida un fichero donde encontrar todos los datos ya ordenados de forma fiable para cada año. Para crear esta función es importante tener en cuenta si faltan datos para algunas horas. Esto se puede corregir utilizando la media de las dos horas vecinas más próximas. También es importante comprobar que estén realizados correctamente los dos cambios horarios anuales, sabiendo que un día del año tendrá 23 horas y otro 25, y también habrá que tener en cuenta si el año es bisiesto o no. Para algunos de los archivos descargados de *Esios*, sucede algo parecido, y habrá que tratar cada fichero de la manera correspondiente.

Todo lo que se acaba de comentar, resulta extremadamente importante a la hora de hacer un estudio de estas características. Es común que los ficheros no vengan siempre de la manera deseada y hay que prestar atención a como se transforma esa información, para que al comparar unas variables con otras, se tenga la seguridad de que se están comparando valores que se dieron justamente en el mismo instante. Los ficheros que se manejan en este estudio pueden llegar a tener más de 35.000 valores horarios consecutivos. Si se relaciona una cantidad importante de valores de unos ficheros con otros pensando que corresponden al mismo instante

y realmente no es así, se pueden llegar a cometer errores muy graves que invaliden el estudio por completo, ya que no estará arrojando un resultado real.

Una vez se tiene la información horaria para todas las variables tanto eléctricas como meteorológicas, se pueden almacenar por columnas en un mismo archivo Excel. A este archivo se le añaden también cuatro columnas más con la información sobre la hora (valores de 1 a 24), día de la semana (valores de 1 a 7), mes (valores de 1 a 12) y año (valores de 2014 a 2017) en la que se encuentran los datos de esa fila. Esto permitirá poder clasificar la totalidad de los datos de manera sencilla, sabiendo qué resultados del PDVP y que condiciones meteorológicas se han dado para cada hora de estos cuatro años. Resultará útil para el caso en que se quieran hacer estudios de horas, días, meses o años concretos.

2.3. ENTRADA DE DATOS A WEKA

WEKA admite la entrada de ficheros de texto con formato *.arff*. Tiene una estructura donde primero se declaran las variables para luego añadir los datos reales asociados a esas variables. En la siguiente imagen (*figura 2.4*), se ve un ejemplo del formato que debe seguir el fichero de texto para que WEKA pueda leer la información.

```
@RELATION EjemploWeka

@ATTRIBUTE DemandaPVP    REAL
@ATTRIBUTE EolicaPVP    REAL
@ATTRIBUTE FotovoltaicaPVP    REAL
@ATTRIBUTE TemperaturaMadrid    REAL
@ATTRIBUTE VientoGibraltar    REAL
@ATTRIBUTE VientoVigo    REAL
@ATTRIBUTE month    REAL
@ATTRIBUTE day    REAL
@ATTRIBUTE hour    REAL
@ATTRIBUTE SentidoDesvioSistema    {1,-1}

@DATA
22588.60000,2624.30000,6.50000,-3.00000,37.10000,7.60000,1.00000
,7.00000,1.00000,-1
20908.50000,2509.40000,6.50000,-3.70000,35.30000,13.00000,1.0000
0,7.00000,2.00000,-1
19706.80000,2354.30000,6.50000,-3.50000,29.50000,9.40000,1.00000
,7.00000,3.00000,-1
18621.30000,2288.50000,6.50000,-3.20000,25.90000,9.40000,1.00000
,7.00000,4.00000,-1
18026.10000,2315.30000,6.50000,-2.10000,24.10000,9.40000,1.00000
,7.00000,5.00000,-1
17888.10000,2111.60000,6.50000,-3.20000,24.10000,5.40000,1.00000
,7.00000,6.00000,-1
```

Figura 2.4 - Formato de entrada de datos a WEKA

Lo primero será añadir una primera línea donde tras la expresión *@RELATION,* se escribirá el nombre que se quiera que reciba el archivo dentro de WEKA. Este nombre es importante que no coincida con el nombre escogido para ningún atributo, lo cual da problemas a la hora de introducir el fichero en el programa. Seguidamente, en el ejemplo, se pretende crear un algoritmo de aprendizaje que ayude a poder predecir el sentido del desvío del sistema conociendo la información relativa a los atributos que en la figura (*figura 2.4*) se observan. Para ello, se irán añadiendo las distintas variables escribiendo sus nombre tras la expresión *@ATTRIBUTE* y añadiendo el tipo de dato que contiene. Para el caso de este estudio serán todos números reales por lo que se debe añadir *REAL* tras el nombre del atributo. El último atributo añadido en el caso del ejemplo es la variable que se quiere predecir, y como solo se encontrarán los valores 1 y -1 en ella (describiendo así el sentido de desvío), se escriben directamente estos dos números entre llaves y separados por coma. A este atributo se le llamará "la clase", ya que será el que clasifique cada conjunto de datos con 1 ó -1.

Por último, tras escribir la expresión *@DATA*, se añadirá la información real que aportan las variables. Esto se hace, añadiendo el primer valor de cada atributo en la primera fila, separados por comas. Luego en la siguiente fila irán los segundos valores de cada atributo y así sucesivamente. Los valores de los atributos de cada fila tienen que estar ordenados del mismo modo que se han ordenado los atributos al declararlos.

Si se incluye toda la información descargada en un archivo Excel, tal como se comentó anteriormente, en los que cada columna contenga la información horaria para cada una de las variables del estudio desde 2014 hasta 2017, es fácil crear una función en Matlab que genere el fichero de entrada a WEKA. Es importante que esa función permita escoger fácilmente los datos que introduce en el fichero si se quieren realizar distintos estudios de manera ágil. Por ejemplo, si se quiere estudiar de manera independiente lo que ocurre en ciertas horas del día o en ciertos días de la semana, al contar con columnas que representan estos datos, será sencillo filtrar y volcar a WEKA solo la información que corresponda a los instantes deseados.

2.4. DEMOSTRACIÓN FUNCIONAL

Hasta aquí, se ha hecho una breve descripción del programa, se ha descrito la información de partida del estudio y se ha explicado el formato con el que introducirla en WEKA. A continuación, se hará una demostración real de las funciones de WEKA. Para ello se utilizará una muestra de los datos pertenecientes a la información de partida del estudio y se realizarán pruebas con ella, navegando por cada una de las funciones que se han utilizado. Se han escogido la totalidad de los atributos descritos en el apartado *2.2.Datos de partida*, pero incluyendo únicamente aquellos valores que pertenecen a días festivos del año 2017.

2.4.1. CARGA DE DATOS

Tal como se explicó anteriormente, tras abrir WEKA, se accede al *Explorer*. Previamente se ha debido de crear el fichero de texto, añadiendo la extensión *.arff* e incluyendo la información en el formato requerido.

Al abrir el Explorer, aparecerá la ventana del *Preprocess* (*figura 2.5*). Lo primero que se debe hacer es cargar el fichero de entrada en la opción *Open File*. Automáticamente se verá que en la columna de la izquierda aparecen todos los atributos que se añadieron al fichero con su nombre. También a la izquierda se verá el número de datos que tiene cada atributo (*Instances*) y el número de atributos (*Attributes*). Para el caso actual se cuenta con la información de 2520 horas y con 34 atributos distintos. De los 34 atributos, uno será el sentido del desvío del sistema. Este atributo será el que se utilice para clasificar el conjunto de datos de cada hora. En la parte superior derecha aparece información estadística sobre los atributos como media, desviación típica y valores máximo y mínimo. *Unique* se refiere al número de valores que sólo aparecen una vez en ese atributo y *Distinct* al número de valores distintos (excluyendo valores repetidos). Un poco más abajo, aparece el desglose de los valores del atributo por clase. En el eje X aparecen los valores del atributo seleccionado en la parte de la izquierda (en la imagen aparece como atributo seleccionado la temperatura en Madrid, aunque podría ser cualquier otro).

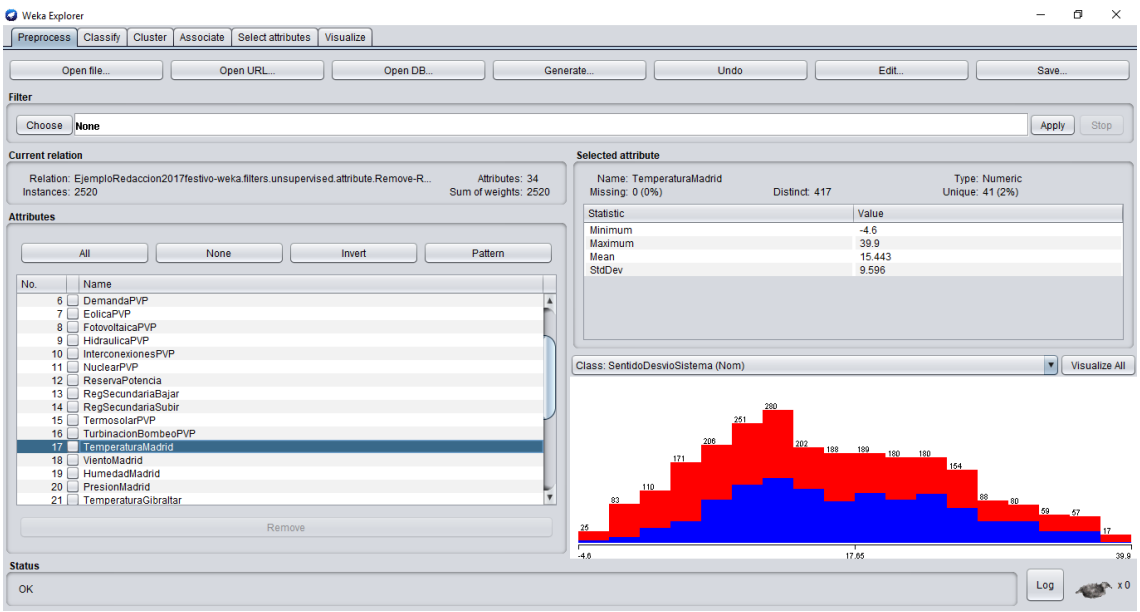


Figura 2.5 - Preprocess con datos cargados

Se recuerda que existen dos posibles casos: desvío a subir del sistema (sobra energía) o desvío a bajar del sistema (necesita energía). Estos dos casos se representan con colores en el gráfico. Se ve para todo el rango de temperaturas, cuantas veces se han dado desvíos a subir (azul) y

cuantas a bajar (rojo). Esta clasificación se hace para cada una de las variables y se pueden ver una a una si se van seleccionando en la columna de la izquierda, pero si se pulsa *Visualize All*, se abre una ventana donde aparece el desglose de todos los atributos a la vez (*figura 2.6*). Esto puede servir para hacer una comprobación rápida de lo útil que resulta cada atributo, considerado por separado.

Figura 2.6 - Ventana de Visualize all

Uno de los objetivos de introducir un número elevado de atributos, es el de encontrar buenos clasificadores, es decir, atributos que sean capaz de separar los datos pertenecientes a distintas clases. Si se recorren uno a uno los gráficos para cada uno de los atributos, es difícil encontrar alguno que marque una frontera clara por sí solo. Por ejemplo, en la imagen anterior donde se observaba la temperatura de Madrid, se veía que a bajas temperaturas, predominaba el color rojo, por lo que, con vistas a realizar una predicción, será más probable que en esos momentos se produzcan desvíos a bajar. Sin embargo no han dejado de producirse desvíos a subir, por lo que se necesitarán de otros atributos y de métodos más complejos para establecer una clasificación fiable.

En la siguiente imagen (*figura 2.7*), se muestra cómo se pueden seleccionar filtros para los datos y los atributos, además de poder borrar algunos de ellos. Los filtros permiten, por ejemplo, normalizar los datos, borrar los que culpan un determinado criterio, crear nuevos atributos, etc. Aunque al venir los datos previamente filtrados desde Matlab, estas herramientas no se utilizarán por el momento.

Figura 2.7 - Ejecución de filtros

2.4.2. CLASIFICACIÓN DE DATOS

Una vez se tienen los datos deseados cargados en el programa, se puede realizar una primera clasificación para los datos. Se pulsa en la pestaña *Classify* que se encuentra justo al lado de *Preprocess* y se abrirá una nueva ventana. En la siguiente imagen (*figura 2.8*) se observa la ventana de *Classify* con indicaciones de sus principales herramientas.

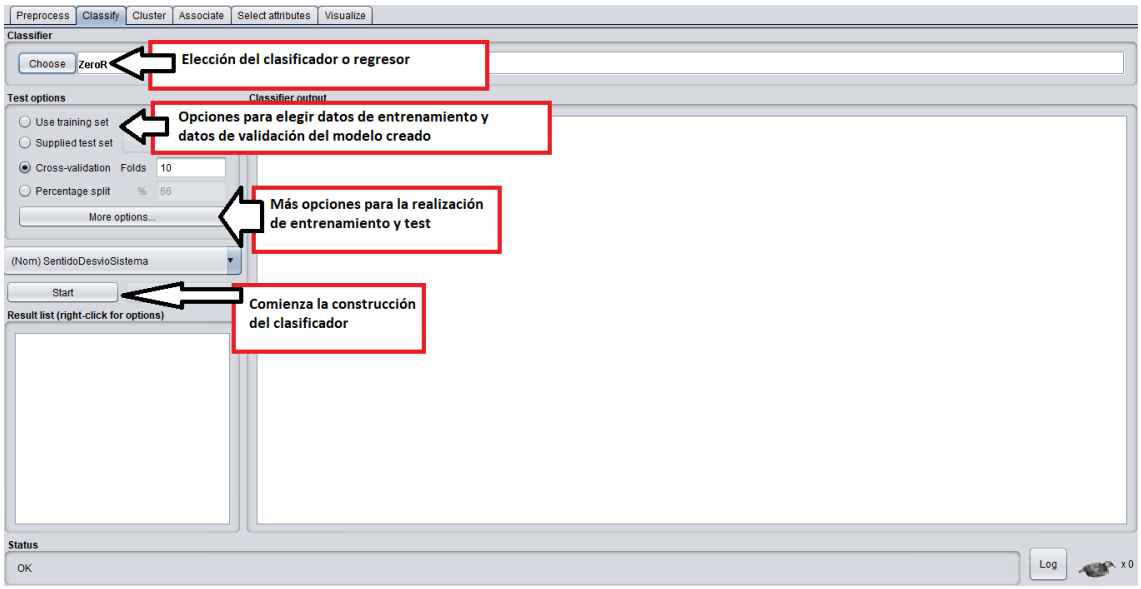


Figura 2.8 - Interfaz del clasificador

Lo primero será elegir el método de predicción pulsando en *Choose*, dónde por defecto aparecerá el clasificador *ZeroR*. Este método, clasifica todos los datos según la clase mayoritaria, es decir, que si en la muestra con la que se va a realizar la prueba, el mayor porcentaje de los desvíos son a subir, clasificará el resto de datos del mismo modo. Es conveniente utilizar primero este método porque dará una pista sobre el porcentaje mínimo de aciertos que habrá que superar con cualquier otro clasificador. Debajo de *Choose*, se encuentran las distintas opciones de test, las cuales son:

- *Use training set*: esta opción utilizará para hacer el test el mismo conjunto de datos que utiliza para hacer el entrenamiento. Con esto se prevé que el modelo creado genere un porcentaje de aciertos demasiado optimista. Al igual que el clasificador *ZeroR* se utilizaba para tener una idea del porcentaje mínimo de aciertos que se debe superar. Esta opción puede utilizarse para tener una idea del máximo que se puede alcanzar.

- *Supplied test set*: si se cuenta con un fichero de datos para testeo distinto del fichero de datos de entrenamiento, es en esta opción donde se podrá seleccionar.

- *Crossvalidation*: esta opción permite hacer una validación cruzada de *K* iteraciones. Los datos de la muestra se dividen en K subconjuntos. En cada una de las *K* iteraciones, el subconjunto *K* se utiliza como datos de prueba y el resto *(K-1)*, se utiliza como datos de entrenamiento. Una vez se termina el proceso de validación cruzada, se realiza la media aritmética de los resultados de cada iteración para obtener un único resultado. Este método resulta preciso puesto que evalúa *K* combinaciones de prueba y entrenamiento aunque desde el punto de vista computacional resulta más lento mientras más iteraciones se añadan. La elección del número de iteraciones debe depender de la medida del conjunto de datos. El programa utiliza por defecto 10 iteraciones.

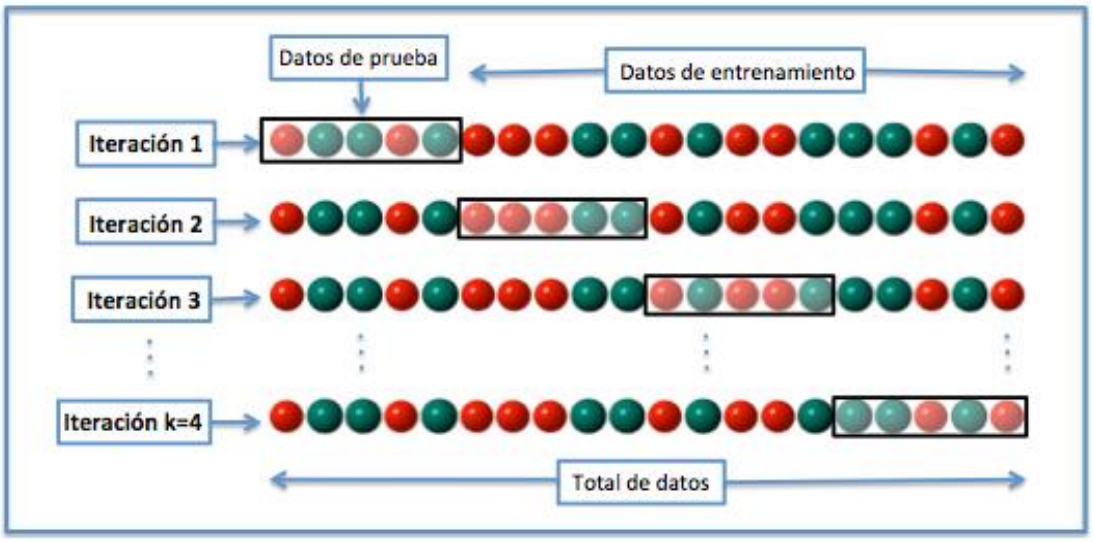


Figura 2.9 - Validación cruzada

- *Percentage split*: para este caso, se puede elegir el porcentaje de la muestra de datos que corresponderá a datos para entrenamiento. Por lo tanto, el porcentaje sobrante corresponderá a datos para testeo. Por defecto aparece 66% con lo que con esta opción se entrenará con el 66% del total y se testeará con el restante 34%. Es importante saber que WEKA desordena aleatoriamente el conjunto inicial y después hace la partición según los porcentajes elegidos. De esta manera, si se construye el clasificador dos veces, se obtendrían desordenaciones distintas y por lo tanto se podrían obtener porcentajes de aciertos en el test ligeramente distintos. En la pestaña *More options*, es posible elegir que no se desordenen los datos seleccionando la opción *Preserve order for % Split*. También existen algunas opciones para elegir la información que aparecerá en la ventana de visualización a la hora de obtener los resultados de la clasificación.

A continuación se va a realizar la clasificación. Por el momento permanece seleccionado *ZeroR*. Como ya se dijo, este método clasifica todos los datos según la clase mayoritaria, si se pulsa *Start*, se realiza la simulación y aparecerán los resultados en la ventana de *Classifier Output*. Se observa (*figura 2.10*) que se obtiene un 54% de aciertos. Si se observa el porcentaje de aciertos desglosado por clase (*TP Rate* o True Positive Rate), se ve que la segunda clase la acierta al 100% (-1,*TP Rate*=1), correspondiendo a la clase de valor -1 o desvío del sistema a bajar. La primera clase la falla al completo (1,*TP Rate*=0). Esto es lógico, dada la manera de funcionar de *ZeroR*, en la que solo acierta la clase mayoritaria. Con esto se sabe que para el conjunto de datos, el 54% corresponderán a ocasiones donde el sistema se desvió a bajar y 46% donde se desvió a subir, con lo que será ese 54% el porcentaje a superar. Más abajo se puede ver la matriz de confusión donde se observa que clasifica todos los datos como desvíos a bajar (*-1*). En cuanto a la matriz de confusión, habrá que procurar que la mayor parte de clasificaciones realizadas se sitúen sobre su diagonal principal, lo que significará que se habrá llegado a un buen número de aciertos.

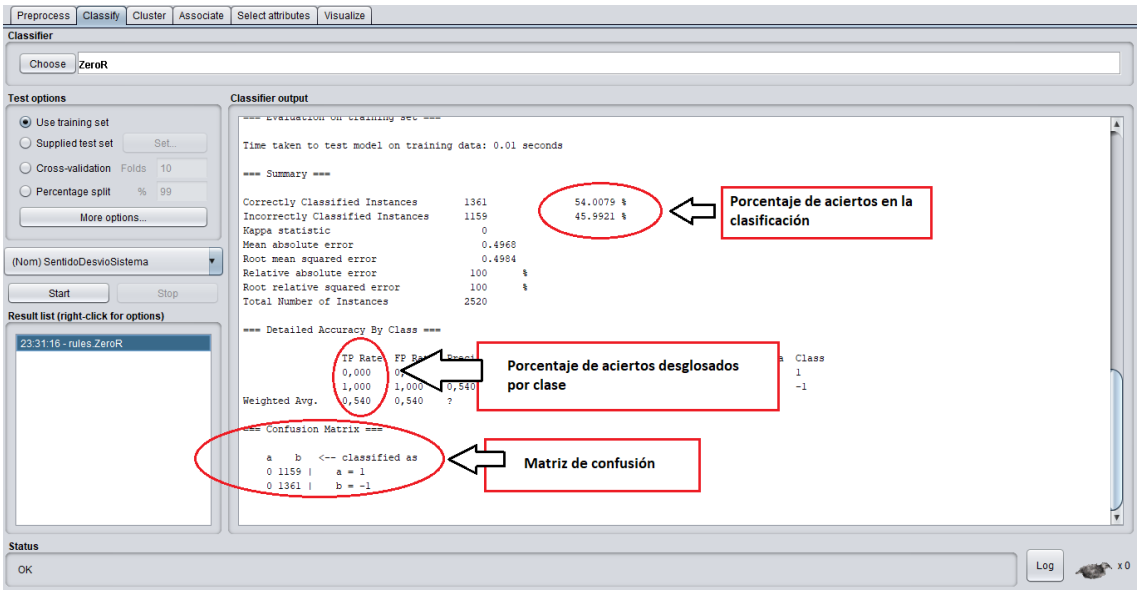


Figura 2.10 - Resultadosde ZeroR

A continuación se va a probar con otro clasificador. Se despliega la lista y aparecen todos los que son posible seleccionar. Para el presente estudio se han elegido las redes neuronales (*MultilayerPerceptron*) y los árboles de decisión (*J48*). Se han elegido ambos métodos porque son buenos realizando clasificaciones de conjuntos de datos, lo cual es lo que se requiere en este estudio: predecir si determinados conjuntos de datos relacionados con una serie de horas, tienen asociado el sentido del desvío a subir o a bajar.

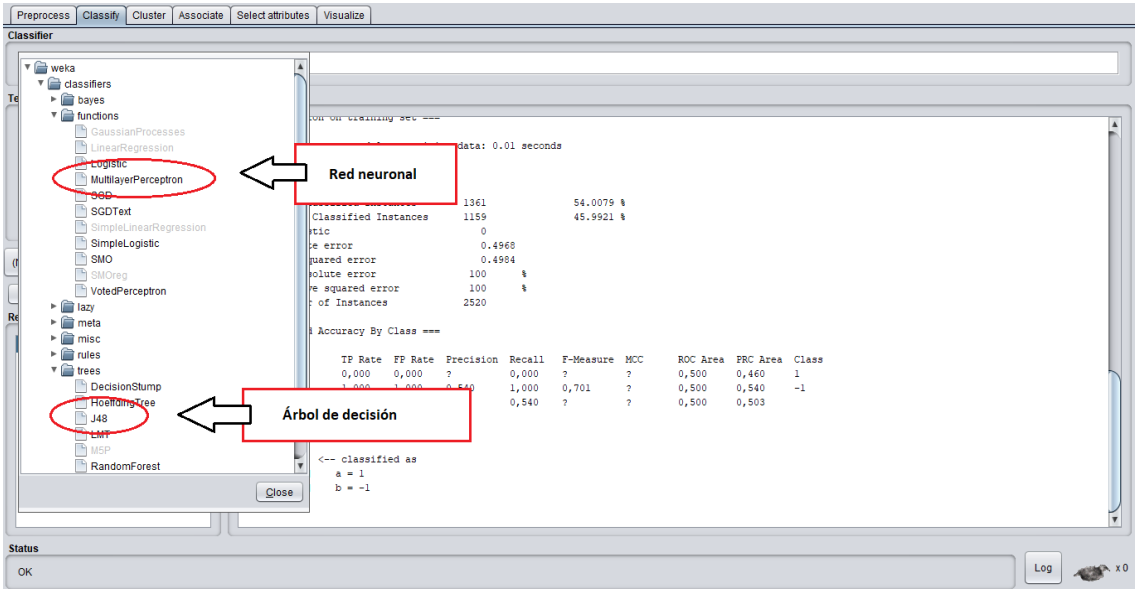


Figura 2.11 - Selección del clasificador

2.4.2.1. ÁRBOL DE DECISIÓN

Primero se realizará la prueba con el árbol de decisión (*J48*). Este clasificador construirá un árbol de decisión partiendo del grupo de datos de entrenamiento usando el concepto de "entropía de información". El tamaño del árbol se mide por el número de nodos y el número de hojas. Se parte desde un nodo inicial, en el que se plantea una condición sobre uno de los atributos y que llevará por un camino u otro hacia otro nodo. Finalmente, nodo tras nodo, se llegará a uno de los puntos finales del árbol (hojas) donde se clasificará el conjunto de datos como de una clase (*1*) u otra (*-1*). En cada uno de los nodos, el árbol elige el atributo que más eficazmente divida al conjunto de muestras en subconjuntos enriquecidos en una clase u otra. El atributo que en cada nodo presente una mayor ganancia de información (o mayor entropía según el concepto que se utiliza), será el elegido como parámetro de decisión para dicho nodo.

Una vez seleccionado, se hará doble click tal como se indica en la imagen (*figura 2.12*) para desplegar la ventana de parámetros del clasificador. Si no se conoce bien el clasificador o la función de alguno de los parámetros, tanto en la pestaña *More* como en *Capabilities*, se podrá obtener más información.

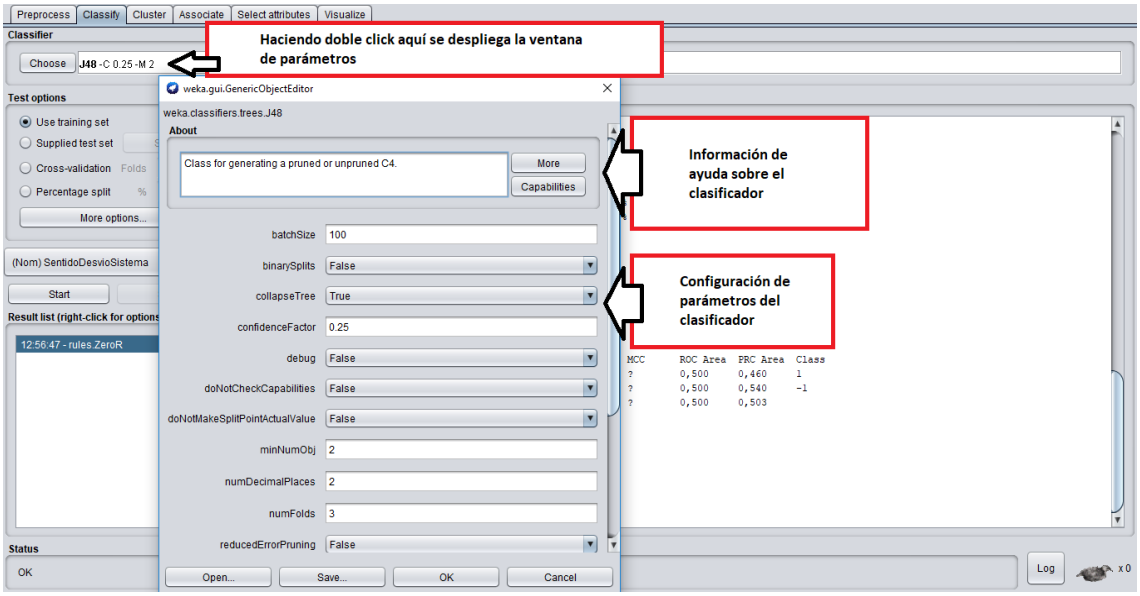


Figura 2.12 - Parámetros del árbol de decisión

Lo primero será acceder a *Capabilities* de donde se extraerá información importante. Tanto en *Class* como en *Attributes* se indica el tipo de dato que acepta este clasificador como atributos y como clase. En los atributos se indica que acepta *Numeric attributes* (entre otros) y en la clase se puede leer *Binary class,* con lo que tanto los atributos (valores numéricos) como la clase escogida (desvío a subir o a bajar) son válidas para este clasificador.

Una vez se conoce que el clasificador es válido para el conjunto de datos que se tiene, es posible extraer más información en *More*. Una vez abierta la ventana, se puede ver en *SYNOPSIS*, un resumen sobre el clasificador que se va a utilizar. Para este caso se lee que el clasificador utiliza un algoritmo, llamado *C4.5*, para generar el árbol de decisión. También aparece una referencia bibliográfica con el nombre del desarrollador del algoritmo (*Ross Quinlan*), por si se desea más información. Esto es lo que ha permitido saber que el clasificador *J48* de WEKA es un método de predicción basado en árboles de decisión.

Si se sigue leyendo, se encuentra una breve explicación de lo que cada parámetro significa. Para este clasificador será muy importante el *confidenceFactor*, el cual define el tamaño del árbol de decisión. No se puede controlar directamente el número de nodos y hojas que tendrá el árbol, pero mientras mayor sea el *confidenceFactor* más complejo tenderá a ser. Este parámetro varía entre 0 y 1, siendo 0.25 el valor por defecto en WEKA.

Una vez se definen los parámetros de la manera deseada (en este caso se dejará la configuración por defecto), se selecciona *Use training set* y se inicia el test, dando un resultado de casi 96% de aciertos. Este porcentaje tan alto se debe a que como ya se ha dicho antes, se entrena y se testea con la muestra completa dando un resultado bastante optimista. Ahora se selecciona *Cross-validation* y se dejan las 10 iteraciones que aparecen por defecto. El resultado ahora baja hasta un 74.5% de aciertos. Ahora será interesante ver como varía el resultado en función de parámetros como el número de iteraciones o el *confidenceFactor*.

Primero se modifica el factor de complejidad bajándolo a 0.1 y luego subiendo a 0.75. Bajándolo a 0.1 se reduce el resultado a 74.4% de aciertos pero el árbol reduce también su complejidad, siendo previamente de 212 hojas y 423 nodos y ahora de 183 hojas y 365 nodos. Ahora, subiendo el factor a 0.75, el porcentaje de aciertos vuelve al anterior 74.5% pero creando un árbol de mayor tamaño (258 hojas y 515 nodos) y con un tiempo de computación mucho mayor. Si para 0.1 y 0.25, la simulación no llega a durar 3 segundos, subiéndolo a 0.75 llega a durar más de 2 minutos. Volviendo al factor de 0.25, se reduce el número de iteraciones a 5, aumentando el porcentaje de aciertos hasta casi 75%. Si se sube a 20, el porcentaje llega a 75.6% y subiéndolo a 40 baja a 74.9%. Todo esto indica que en cuanto a relación resultados y tiempo de computación, los valores predefinidos por WEKA son los más aconsejables. En la ventana de resultados vendrá dado el tamaño y el modelo del primer árbol creado de todas las iteraciones. Si se quieren visualizar todos, será necesario marca la casilla *Output models for training splits* en la ventana de *More options*.

Ahora se hará la simulación eligiendo un porcentaje de entrenamiento y otro de testeo. Escogiendo 66% para el entrenamiento, tal como viene definido, se alcanza un 72.3% de aciertos. Si se reduce el porcentaje de entrenamiento, los aciertos solo hacen reducirse, pero si por el contrario se aumenta hasta un 99%, se obtiene un 84% de aciertos. En este último caso el tamaño de la muestra se ha reducido a 25 grupos de datos a clasificar, prácticamente como si se hubieran querido predecir las 24 horas de un día partiendo de la información de 100 días anteriores. Estas divisiones entre muestra de entrenamiento y muestra de testeo se han hecho partiendo de una pila de datos desordenada, si se mantiene el orden en el caso de 66%, el porcentaje de aciertos cae hasta 54.3% y en el caso de la división del 99%, cae hasta un 76%. Para ordenar o desordenar los datos habrá que marcar la casilla de *Preserve order for % Split,* tal como se indicó anteriormente.

Tal como se van haciendo simulaciones, estas van almacenando sus resultados en la columna inferior izquierda (*Result list*) y se pueden consultar aunque se simulen otras nuevas. Si se hace click derecho sobre alguna de las simulaciones realizadas, aparecen varias opciones, como son guardar el modelo creado, visualizar el árbol creado y otra serie de visualizaciones o de operaciones disponibles. Si se desea, se puede ver gráficamente como sería el árbol creado (*figura 2.13*), aunque si lo que se quiere es tener una visión general, las dimensiones hacen que no se puedan distinguir los nodos, pero es posible hacer zoom e ir reconociendo partes concretas tal como también se muestra en la imagen.

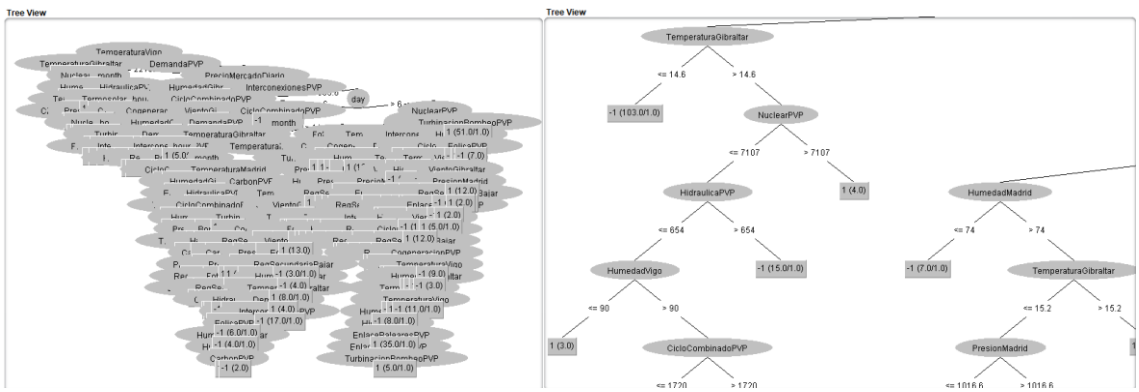


Figura 2.13 - Árbol completo y detalle

Por otro lado, también es importante prestar atención a la matriz de confusión que se muestra en los resultados, porque puede aportar información de cómo mejorar la predicción. Si se observan los resultados de la simulación donde se ha elegido 99% de la muestra para entrenar, se consiguió un 84% de aciertos, acertando en la clasificación de 21 datos y fallando en la de 4. La matriz de confusión indicaba que se erraba clasificando 3 datos como desvíos a subir y 1 como desvío a bajar, es decir, que se cometen más errores clasificando uno de los datos que con el otro. Pudiese ser también que se tuviese conocimiento de que los desvíos en un sentido fuesen más caros que en el otro y que en caso de duda interesase que el algoritmo clasificase con el desvío más barato en caso de duda. Esto es posible hacerlo con un metaclasificador.

```
Time taken to test model on test split: 0 seconds

=== Summary ===

Correctly Classified Instances          21               84      %
Incorrectly Classified Instances         4               16      %
Kappa statistic                          0.6815
Mean absolute error                      0.1767
Root mean squared error                  0.3897
Relative absolute error                 35.6866 %
Root relative squared error             78.4371 %
Total Number of Instances               25

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
               0,909     0,214     0,769       0,909   0,833      0,690    0,821     0,700     1
               0,786     0,091     0,917       0,786   0,846      0,690    0,821     0,842     -1
Weighted Avg.  0,840     0,145     0,852       0,840   0,841      0,690    0,821     0,779

=== Confusion Matrix ===

  a  b   <-- classified as
 10  1 |  a = 1
  3 11 |  b = -1
```

Figura 2.14 - Resultados sin alterar la matriz de costes

Si se va a la ventana donde se seleccionan los clasificadores, se despliega la pestaña *Meta* y se escoge el *CostSensitiveClassifier*. Este clasificador puede alojar el algoritmo de otro, como puede ser el *J48* que se ha utilizado hasta ahora, y modificar los costes de la matriz de confusión. En la imagen (*figura 2.15*) aparece desplegada la ventana de configuración de este metaclasificador, donde habrá que seleccionar *J48* en *classifier* y hacer doble click en *costMatrix*. Esto abrirá una nueva ventana donde en *Class*, se debe poner el número de clases que existen en el problema, en este caso se pondrá 2. Automáticamente aparecerá una matriz con 0 en su diagonal y 1 en el resto de posiciones. Esto es así porque clasificar datos en la diagonal tiene coste cero (esto significa que se ha acertado en la predicción). Pero en cambio clasificar datos de una clase como de la clase contraria sí tiene un coste, en este caso 1.

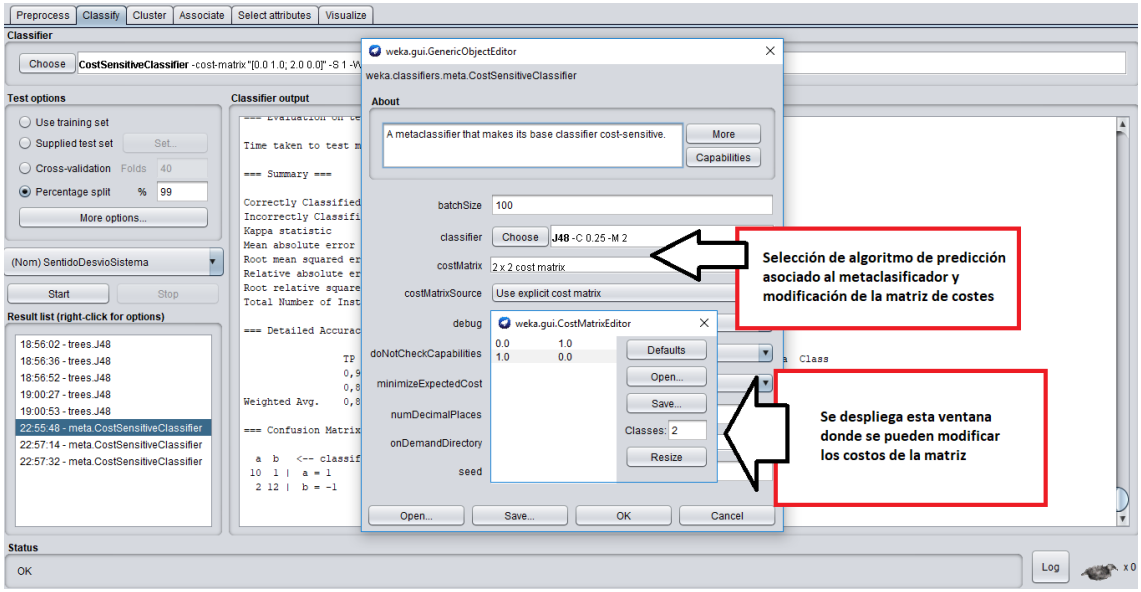


Figura 2.15 - Configuración del metaclasificador

Dado que se realizan incorrectamente 3 clasificaciones como desvíos a subir y 1 como desvíos a bajar, se va a modificar el coste de la clasificación que más error comete. Para ello, se aumenta el coste de 1 a 1.2 en la posición correspondiente a donde había un 3 en la matriz de confusión. Se cierra y se vuelve a realizar la simulación obteniendo el siguiente resultado (*figura 2.16*).

```
Time taken to test model on test split: 0 seconds

=== Summary ===

Correctly Classified Instances          22               88      %
Incorrectly Classified Instances         3               12      %
Kappa statistic                          0.7588
Mean absolute error                      0.1627
Root mean squared error                  0.3536
Relative absolute error                 32.8475 %
Root relative squared error             71.1834 %
Total Number of Instances               25

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,909    0,143    0,833      0,909    0,870      0,761    0,812     0,696     1
                 0,857    0,091    0,923      0,857    0,889      0,761    0,812     0,821     -1
Weighted Avg.    0,880    0,114    0,884      0,880    0,880      0,761    0,812     0,766

=== Confusion Matrix ===

  a  b   <-- classified as
 10  1 |  a = 1
  2 12 |  b = -1
```

Figura 2.16 - Resultado modificando matriz de costes

Como se comprueba, el modificar la matriz de costes de la manera indicada, ha supuesto una mejora en los resultados de la predicción, subiendo de un 84% a un 88% de acierto. Aún se siguen cometiendo dos errores clasificando erróneamente como desvío a subir y uno solo como desvío a bajar. Pero si se sigue aumentando el coste de clasificar como desvío a subir, se mejorará este indicador a costa de empeorar el otro, con lo cual, no tiene sentido continuar modificando la matriz de costes.

2.4.2.2. PERCEPTRÓN MULTICAPA

Ahora se cambiará de algoritmo de clasificación y se realizará la misma tarea que con el árbol de decisión. Si se despliega de nuevo la lista de clasificadores y se abre la pestaña *functions*, aparecerá el clasificador *MultilayerPerceptron*. Este algoritmo se encuentra dentro de la clasificación de red neuronal. Su arquitectura se caracteriza básicamente porque tiene sus neuronas agrupadas en capas de distintos niveles. Cada una de las capas está formada por un conjunto de neuronas y se distinguen tres tipos de capas distintas: la capa de entrada, la capa oculta y la capa de salida.

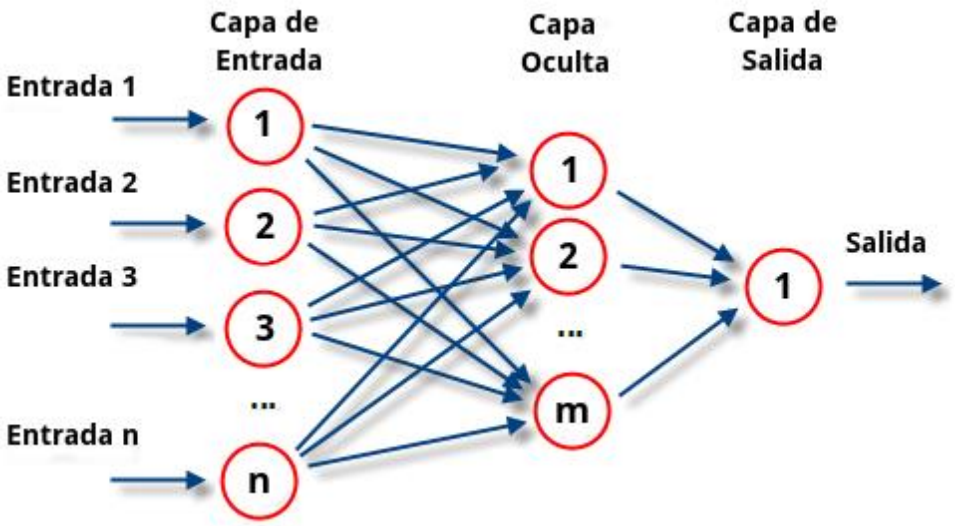


Figura 2.17 - Estructura del perceptrón multicapa

Las neuronas de la capa de entrada no actúan como neuronas propiamente dichas, sino que se encargan únicamente de recibir las señales o patrones del exterior (en este caso serán los atributos) y propagar dichas señales a todas las neuronas de la siguiente capa. La última capa actúa como salida de la red, proporcionando al exterior la respuesta de la red para cada uno de los patrones de entrada. Las neuronas de las capa oculta o capas ocultas (pueden ser más de una), realizan un procesamiento no lineal de los patrones recibidos. Esto lo hacen definiendo una serie de pesos para cada una de sus conexiones, los cuales hacen que al entrar una serie de datos de entrada, vayan tomando un camino hasta llegar al valor de clasificación.

Si se selecciona el clasificador y se abre su ventana de configuración, se puede pulsar *Capabilites* y ver si es apto para tratar los datos de entrada y salida. En la clase y en los atributos se puede ver respectivamente *Binary class* y *Numeric attributes* entre otros. Por lo tanto es un algoritmo apto para clasificar los datos de entrada. Si se pulsa en *More*, se podrá ver el resumen que hace WEKA sobre este clasificador. En este caso, se explica que el Perceptrón Multicapa utiliza retropropagación para realizar el entrenamiento de las neuronas.

En la retropropagación, se aplica un patrón a la entrada como estímulo (entrada de datos de los atributos) y este se propaga desde la primera capa hasta las siguientes, generando una salida. La señal de salida se compara con la salida deseada y se calcula una señal de error para cada una de ellas. Las señales de error se propagan hacia atrás, partiendo de la capa de salida, hacia todas las neuronas de la capa oculta que contribuyen directamente a esa salida. Sin embargo las neuronas de la capa oculta solo reciben una fracción de la señal total del error, basándose aproximadamente en la contribución relativa que haya aportado cada neurona a la salida original. Este proceso se acaba repitiendo para todas las capas hasta que todas las neuronas hayan recibido una señal de error que describa su contribución relativa al error total. Posteriormente, durante la fase de testeo, cuando a las neuronas se les presente un patrón arbitrario que contenga ruido o que esté incompleto, las neuronas de la capa oculta responderán con una salida activa si la nueva entrada contiene un patrón que se asemeje a aquella característica que las neuronas individuales hayan aprendido a reconocer durante su entrenamiento.

Si se observan los distintos parámetros, hay algunos que resultan interesantes, en concreto hay tres que influyen en gran medida en la complejidad y funcionamiento de la red: *hiddenLayers*, *learningRate* y *momentum*. El primero define número de capas ocultas y número de neuronas en cada capa que tendrá la red. El segundo es el encargado de controlar cuanto cambian los pesos de la red en función del error obtenido y el tercero aporta estabilidad al segundo cuando este es tan alto que provoca divergencias en el modelo.

Para el número de neuronas en la capa oculta, viene por defecto la siguiente fórmula:

$$NeuronasCapaOculta = \frac{(Nº\ atributos + Nº\ clases)}{2} \qquad (Ecuación\ 1)$$

Lo cual WEKA lo codifica con la letra ‘a’. También se podrán elegir el número de neuronas en la capa oculta mediante otras letras: ‘i’,’o’ y ‘t’, representando estas a: el número de atributos, el número de clases o el número de clases más el de atributos, respectivamente. Independientemente se puede escoger el número de neuronas que se desee, lo anterior son solo opciones predefinidas que da el programa. Es posible crear más de una capa oculta. Solo será necesario escribir *n* valores separados por comas y se crearán *n* capas ocultas del número de neuronas especificado por cada uno de los valores. En este estudio se utilizará una sola capa.

En cuanto a factor de aprendizaje (*learningRate*) y a la “cantidad de movimiento” (*momentum*), WEKA da 0.3 y 0.2 como valores por defecto. En principio las simulaciones se harán con todos

los valores por defecto y luego se verá cómo cambian algunos de los resultados en función de estos parámetros.

También existen otros parámetros importantes en la configuración del Perceptrón Multicapa. Para este algoritmo es posible seleccionar que una parte del conjunto de entrenamiento sirva para validar el modelo a medida que se va iterando y poder parar esta iteración si se detecta que se ha llegado a un mínimo de la señal de error o se están cometiendo demasiados errores consecutivos en cada iteración. Estos parámetros son *ValidationSetSize, ValidationThreshold*¸ y *TrainingTime*. El primero fija un porcentaje de la muestra de entrenamiento que podrá ser utilizada para validar el modelo. El segundo establece el número de veces seguidas que se permitirá que aumente el error obtenido del proceso de validación tras cada iteración antes de parar el entrenamiento. Por último, el tercero establece el número máximo de iteraciones que se realizarán en el entrenamiento. Si no se selecciona un porcentaje para la muestra de validación, se realizarán todas las iteraciones que marque este parámetro, pero de haberse fijado un porcentaje, el proceso iterativo podrá pararse antes. Una iteración completa consistirá en la introducción de todas las filas de datos del conjunto de entrenamiento una vez, habiendo modificado los pesos en función de las salidas obtenidas. El programa establece de inicio un porcentaje nulo para el conjunto de validación y un *TrainingTime* de 500 iteraciones. Estos valores se dejarán por defecto con lo que no habrá muestra de validación.

En principio, se recuerda que el resultado con el clasificador *ZeroR* fue de 54% de aciertos, con lo que esto será lo mínimo que se deba aceptar. La primera simulación se hace utilizando la opción *Use training set,* que ya se vio que realizaba el test con la misma muestra que usaba para entrenar el modelo. Lo primero que se observa es que el tiempo de computación es bastante mayor en comparación con el *J48.* El porcentaje de aciertos ha sido de 90.5%. Comparando con el *J48*, se reduce el porcentaje ideal de aciertos de 96 a 90.5%. Esto ha sido para un número de neuronas de la capa oculta igual al que venía por defecto (*Ecuación 1*.). Teniendo en cuenta que se tienen 33 atributos y dos clases, se habrán cogido 17 o 18 neuronas, según hacia donde haya redondeado el programa. Ahora se prueba a aumentar el número de neuronas al máximo valor predefinido por el programa, que será el de la suma de atributos y clases (35). Se obtiene un 97.6% de aciertos pero un tiempo de computación de 48 segundos. Esto resulta casi el doble de utilizar el anterior valor de neuronas (25 segundos). Si se coge el mínimo predefinido de neuronas, que corresponde con el número de clases (2), se obtiene un 75% con casi 4 segundos de tiempo de computación. Resulta muy rápido pero para ser una prueba realizada sobre la misma muestra de entrenamiento no resulta mínimamente aceptable.

Se vuelve a poner el mismo valor de neuronas que venía por defecto, se selecciona la validación cruzada con 10 divisiones y se pulsa *Start*. Se obtiene 76.5% de aciertos en un tiempo de computación de 24 segundos por modelo, teniendo en cuenta que ha calculado 10 modelos, este tiempo se dispara a 4 minutos. Subiendo el número de neuronas a 35, el tiempo de computación sube a 46 segundos por modelo, lo que hacen cerca de 8 minutos, subiendo a casi un 78.2% de aciertos. Si en la configuración se suben ligeramente ahora los parámetros de *learningRate* y *momentum*, se vuelve a simular y se obtiene un 78.7% de aciertos aunque aumentando el tiempo de computación a 48 segundos por modelo creado. Por último, antes de terminar con la validación cruzada se hace una simulación con la configuración de parámetros inicial pero con 5 divisiones, lo que reducirá el tiempo a la mitad. Se obtiene un 76.4%

disminuyendo ligeramente el porcentaje que se obtuvo con 10 divisiones. No se hará simulación para 20 divisiones porque lo único que se puede esperar es que suba ligeramente el porcentaje de aciertos a un costo computacional muy elevado. Por el momento, se empieza a ver que con la red neuronal pueden conseguirse, ligeramente, mejores resultados que con el árbol de decisión pero con un tiempo de computación bastante más elevado, ya que el *J48* da resultados casi instantáneos. Esto sobre todo resulta tedioso para realizar pruebas cambiando distintos parámetros y haciendo más compleja la red o aumentando el número de atributos. Aunque si se tienen directrices claras de cómo llegar a un modelo bien definido, puede merecer la pena dedicar algunos minutos más de simulación para conseguir un mejor resultado.

Ahora se pasa a realizar una sola división entre muestra de entrenamiento y de testeo. Se comienza con el 66% que aparece por defecto y con la configuración habitual de parámetros. El porcentaje de aciertos resulta de casi 77%, lo cual es de los mejores porcentajes conseguidos hasta ahora en un tiempo de 24 segundos. Aunque este porcentaje se ve drásticamente reducido a 52% si se impone orden en la muestra.

Si se aumenta la división al 99% de la muestra como se hizo anteriormente, los aciertos suben al 84% en el caso de muestra desordenada, y bajan a 44% en el caso de muestra ordenada. Subiendo el número de neuronas ocultas a 35, el resultado con la muestra ordenada sube a 80%, lo cual ya empieza a ser bastante interesante. Una vez visto que el algoritmo funciona mejor aumentando el número de neuronas ocultas, se van a hacer pruebas modificando el *learningRate* y el *momentum*. Si se aumenta el primero de 0.3 a 0.5, se obtiene un porcentaje de aciertos del 88%, con un tiempo de 46 segundos. Viendo que al aumentar este parámetro, mejora la predicción, se sube a 0.6 y se vuelve a simular, bajando el acierto hasta el 72%. Ahora es cuando se entra a modificar el *momentum*, subiéndolo de 0.2 a 0.3, manteniendo el *learningRate* en 0.6. Se obtiene 92% de aciertos, lo cual supone que se han acertado 23 de 25 horas. Al haber utilizado además, una muestra de datos ordenada, resulta un caso bastante real de lo que se podría llegar a hacer: tratar de predecir las 24 horas siguientes en función de los datos de los últimos 100 días.

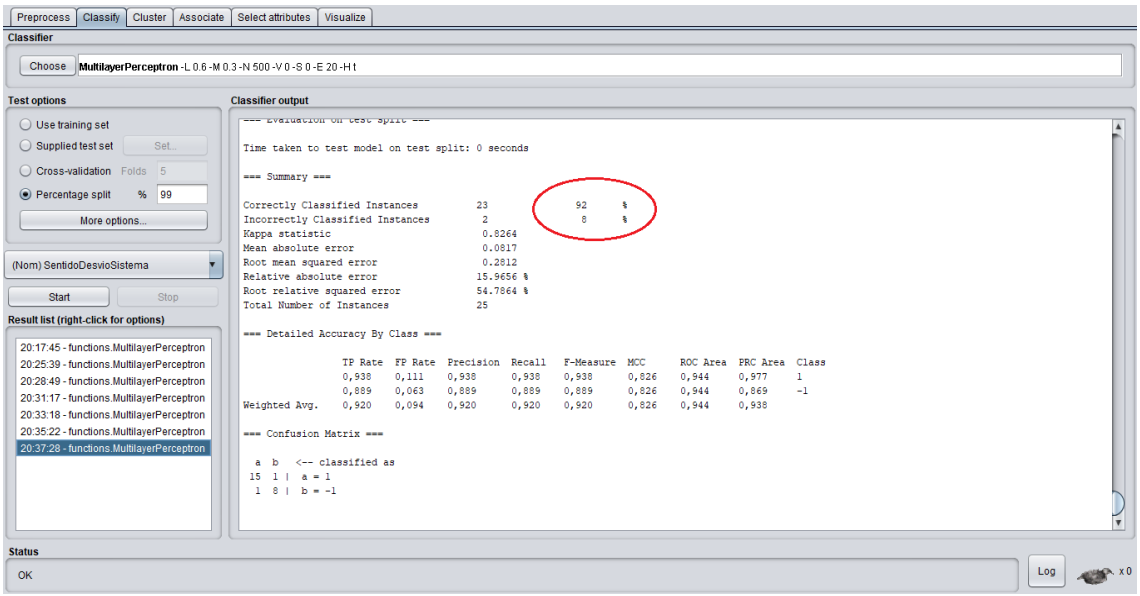


Figura 2.18 - Resultado de 92% de acierto con MultilayerPerceptron

Se siguen probando otras combinaciones de parámetros pero esta última es la que ofrece mejores resultados. Si se selecciona previamente en *More options*, la opción de *Output predictions* y se escoge la opción de *PlainText*, aparecerá la predicción realizada junto con los valores reales y el error (*figura 2.19*).

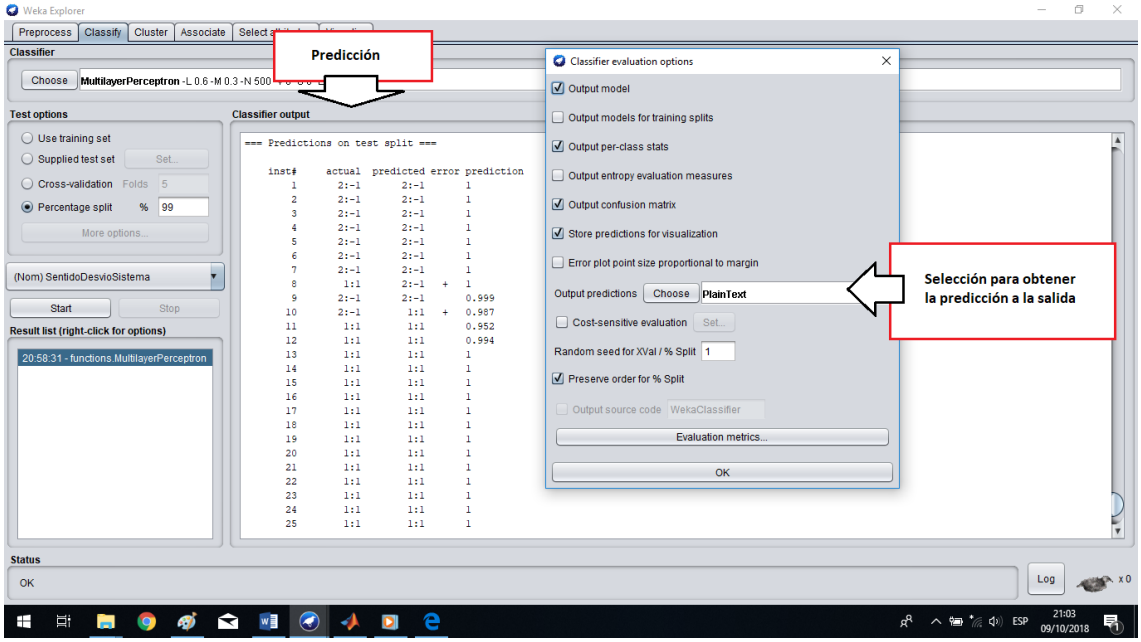


Figura 2.19 - Predicción en la salida

Si se hace click derecho en la simulación que ha generado el resultado más favorable (en *Result list*), se despliegan una serie de opciones. Una de ellas es *save model*, que permitirá guardar el modelo generado para poderlo aplicar a futuros datos. Este modelo se ha generado con la posibilidad de testearlo posteriormente con los últimos datos sobre el sentido del desvío, pero en una situación real, se hubieran tenido los valores de los atributos para las últimas 25 horas pero no el sentido del desvío. Por lo tanto, el primer paso del procedimiento es realizar una predicción usando la opción *Use training set* pero sin contar con los valores de las últimas 25 horas. Una vez hecho esto, se guarda el modelo generado y se aplica sobre el conjunto de 25 datos para cada atributo que ya se conoce para el día siguiente.

Para hacer esto mediante WEKA, lo más rápido e intuitivo es generar para los datos de las 25 horas del día siguiente, un fichero *.arff* igual que los anteriores, pero al no saber cuál será el sentido del desvío, se coloca un 1 por defecto a todas las horas. Este valor no será importante, ya que aplicando el modelo al conjunto de datos, el programa hará su predicción independientemente del resultado que lea como correcto. Para aplicar el modelo guardado a un conjunto de datos, se deberá cargar primero el conjunto de datos en la opción *Supplied test set*. Luego se hace click derecho en la ventana de *Result list* (aunque la ventana esté completamente en blanco, se abrirá una lista desplegable) y se selecciona *Load model*. Con esto, aparecerá un nuevo resultado en la lista que hace referencia al modelo cargado. Se hace click derecho sobre él y se selecciona *Re-evaluate model on current test set*. Con esto se estará aplicando el modelo a este último conjunto de datos, obteniéndose en este caso un 68% de

aciertos. Pero no es este resultado lo que interesa porque ya se sabe que es falso. Si previamente se ha seleccionado la aparición de la predicción con los resultados, se habrá generado en la ventana, un vector con la predicción del sentido del desvío para cada una de las horas del día siguiente. Este vector de 25 valores es el que realmente interesa, ya que, los 25 atributos se clasificaron con clase 1 por defecto simplemente para que pudiesen entrar al programa. Una vez obtenido este resultado, se podrá utilizar para tomar las decisiones que se consideren oportunas a la hora de ofertar.

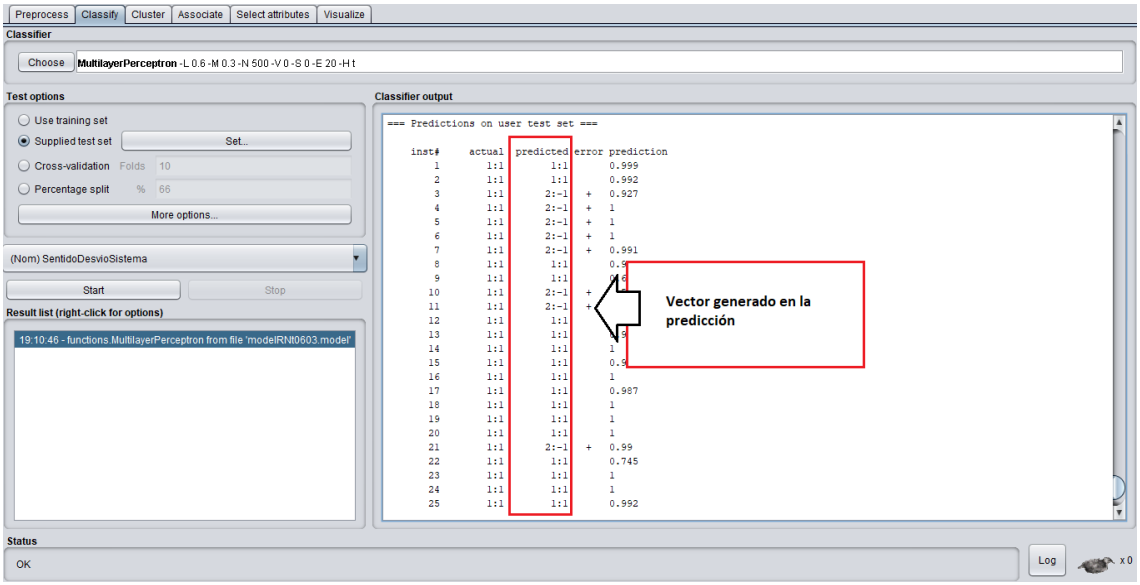


Figura 2.20 - Predicción de 25 horas

Hasta ahora, se ha visto de manera general la funcionalidad de la herramienta de clasificación para los dos algoritmos escogidos. Se ha visto que con árboles de decisión se generaban modelos de forma muy rápida pero daban peores resultados que con redes neuronales. Aun así, los resultados no llegan a ser malos porque se supera en buena medida al clasificador *ZeroR*. Con redes neuronales se alcanzan predicciones bastante más precisas pero tienen el inconveniente del tiempo de computación. Aunque esto solo será a la hora de generar modelos. Una vez se tiene un modelo que se considere fiable, el testeo se hace de manera instantánea. También hay que tener en cuenta la cantidad de datos que se manejan: 33 atributos con 2520 valores cada uno. Desde ese punto de vista, se puede entender que uno de los métodos tarde algo más en dar un buen resultado y otro tarde muy poco en dar un resultado peor pero que ya aporta más información de la que se tenía previamente. Ahora, para ambos algoritmos, se tratará de acotar el problema realizando una selección de atributos. Con ello se puede esperar una mejora de resultados y un menor tiempo de computación.

2.4.3. SELECCIÓN DE ATRIBUTOS

Para acceder a la selección de atributos, habrá que ir a la pestaña de *Select attributes*. Ahora habrá que tomar dos decisiones para poder realizar la selección. La primera, será elegir el método de evaluación de los atributos y la segunda, el método de búsqueda. Hay dos métodos de búsqueda principales, los cuales son: *Ranker* y *GreedyStepwise*. Al igual que con los distintos clasificadores era posible ver el funcionamiento de cada uno y la explicación de cada uno de sus parámetros, con los métodos de búsqueda y de evaluación también será posible obtener esta información. El método *Ranker,* básicamente ordena los distintos atributos utilizando el criterio del método de evaluación que se haya elegido para decidir si cada uno por separado, son mejores o peores para realizar una predicción. Con *GreedyStepwise* se utiliza el método de evaluación para encontrar un grupo de atributos que en conjunto den un buen resultado. Por otro lado, para cada uno de los métodos de evaluación, habrá uno de estos dos métodos de búsqueda asociado. No habrá ningún método que pueda funcionar de igual manera con los dos.

Con esto, existen tres posibilidades principales a la hora de clasificar atributos:

- Evaluación independiente de atributos. Utilizando *Ranker* y por ejemplo un método de evaluación afín como puede ser *InfoGainAttributeEval*.

- Evaluación de conjuntos de atributos. Utilizando *GreedyStepwise* y un método de evaluación afín, para lo cual se tienen dos opciones:

    - Utilizar un método que directamente filtre los atributos y proporcione un conjunto de ellos.

    - Utilizar un método *Wrapper*, el cual requiere que se seleccione un clasificador base para realizar la selección de atributos. Este clasificador puede ser la red neuronal o el árbol de decisión utilizado previamente.

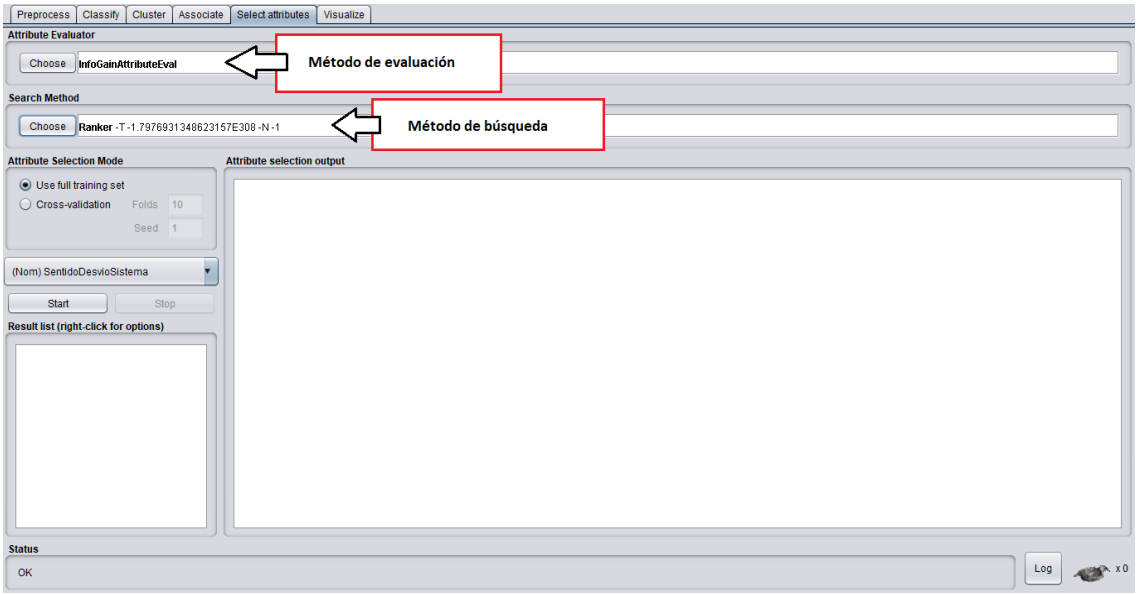


Figura 2.21 - Ventana de selección de atributos

Se empezará escogiendo *InfoGainAttributeEval*, el cual utiliza *Ranker* como método de búsqueda. Este método de evaluación, evalúa cada atributo, midiendo su ganancia respecto de la clase (el sentido del desvío del sistema en este caso). Antes de pulsar *Start*, habrá que escoger un modo de selección de atributos, entre *Use full training set* y *Cross-validation*. Ambas son opciones que ya se conocen y básicamente, con uno se obtiene una sola evaluación para cada atributo y con la otra, una media de las evaluaciones que se hayan hecho según número de divisiones de la muestra. Se hará una simulación con cada una de las opciones y en concreto para la segunda, se escogerán 4 divisiones. Se reduce el número de divisiones pretendiendo disminuir el tiempo de cálculo, ya que solo se quiere hacer una demostración de los resultados de salida.

Use full training set

```
Ranked attributes:
 0.07754   11 NuclearPVP
 0.05595   29 month
 0.03837   21 TemperaturaGibraltar
 0.03445   17 TemperaturaMadrid
 0.0288    31 hour
 0.02675   25 TemperaturaVigo
 0.02319   10 InterconexionesPVP
 0.02302   15 TermosolarPVP
 0.02286    8 FotovoltaicaPVP
 0.02271    6 DemandaPVP
 0.01506   28 PresionVigo
 0.01506   27 HumedadVigo
 0.01406   32 PrecioMercadoDiario
 0.01384    1 EnlaceBalearesPVP
 0.0138    22 VientoGibraltar
 0.01241    3 CarbonPVP
 0.01238   24 PresionGibraltar
 0.01181   16 TurbinacionBombeoPVP
 0.01025    5 CogeneracionPVP
 0.00931   19 HumedadMadrid
 0.0085    13 RegSecundariaBajar
 0.00824   20 PresionMadrid
 0.00698    2 BombeoPVP
 0.00526   14 RegSecundariaSubir
 0.00458   30 day
 0.00441   26 VientoVigo
 0          4 CicloCombinadoPVP
 0         18 VientoMadrid
 0          7 EolicaPVP
 0         23 HumedadGibraltar
 0         12 ReservaPotencia
 0          9 HidraulicaPVP
```

Cross-validation 4 folds

```
average merit      average rank  attribute
 0.056 +- 0.002      1.3 +- 0.43    29 month
 0.034 +- 0.004      3.3 +- 1.09    17 TemperaturaMadrid
 0.03  +- 0.005      4   +- 1.22    31 hour
 0.03  +- 0.002      4.3 +- 0.43    21 TemperaturaGibraltar
 0.029 +- 0.004      5.8 +- 1.64    25 TemperaturaVigo
 0.025 +- 0.006      7.3 +- 2.68     6 DemandaPVP
 0.041 +- 0.024      7.3 +- 7.6     11 NuclearPVP
 0.023 +- 0.005      9   +- 2.12     8 FotovoltaicaPVP
 0.021 +- 0.004      9.8 +- 2.05    15 TermosolarPVP
 0.02  +- 0.005     10.5 +- 2.06    10 InterconexionesPVP
 0.015 +- 0.003     13   +- 1.58     3 CarbonPVP
 0.014 +- 0.001     14   +- 1.87    22 VientoGibraltar
 0.014 +- 0.009     14.3 +- 7.12    28 PresionVigo
 0.014 +- 0.003     14.5 +- 2.29     1 EnlaceBalearesPVP
 0.012 +- 0.003     15.5 +- 3.35    16 TurbinacionBombeoPVP
 0.014 +- 0.009     15.5 +- 9.94    27 HumedadVigo
 0.011 +- 0.001     17.3 +- 1.3      5 CogeneracionPVP
 0.009 +- 0.006     18.5 +- 6.34    24 PresionGibraltar
 0.009 +- 0.001     18.5 +- 1.66    19 HumedadMadrid
 0.01  +- 0.002     18.5 +- 2.29    13 RegSecundariaBajar
 0.009 +- 0.002     19.5 +- 2.29    20 PresionMadrid
 0.005 +- 0.001     22   +- 1.22    14 RegSecundariaSubir
 0.005 +- 0.001     22.8 +- 1.09    30 day
 0.002 +- 0.004     23.8 +- 2.28     2 BombeoPVP
 0.002 +- 0.003     24   +- 2.35    26 VientoVigo
 0     +- 0         24.8 +- 0.83     4 CicloCombinadoPVP
 0.005 +- 0.009     26   +-10.39    32 PrecioMercadoDiario
 0.002 +- 0.003     27   +- 3.54    18 VientoMadrid
 0     +- 0         27.3 +- 0.43     7 EolicaPVP
 0     +- 0         28.8 +- 0.43    12 ReservaPotencia
 0     +- 0         29.8 +- 1.09    23 HumedadGibraltar
 0     +- 0         30.8 +- 0.43     9 HidraulicaPVP
```

Figura 2.22 - Selección con método Ranker

En la imagen anterior (*figura 2.22*), se observan dos columnas distintas. La primera corresponde al uso de *Use full training set*, con la cual, al utilizar una única muestra, clasifica cada atributo con un único número, medido con *InfoGainAttributeEval*. En la segunda columna se dan dos informaciones: el *average merit* y el *average Rank*, ambos con sus desviaciones típicas. Con *average merit*, se refiere a la media de las correlaciones obtenidas en cada uno de los cuatro

ciclos de validación cruzada, lo cual clasifica cada atributo como mejor o peor dentro del ranking. La columna de *average Rank*, se refiere al orden medio en el que quedó cada atributo en cada uno de los cuatro ciclos. Por ejemplo, para el primero de los atributos (*month*), el orden medio es de 1.3 +- 0.43, lo que indica que probablemente debió quedar a veces primero y a veces segundo en la clasificación. Resulta también curioso como el resultado de la nuclear queda primero en la clasificación de muestra única, pero cae al séptimo puesto cuando se divide en cuatro, con un orden medio de 7.3 +-7.6. Esto da la pista de que con toda probabilidad quedó primera en algunas de las divisiones, pero que una mala clasificación lo ha bajado 6 posiciones. Ahora, si se quisiese realizar una selección de atributos para una posterior clasificación, se debería marcar una frontera entre los atributos que se tendrán en cuenta y los que no. Para este caso, parece que 10 es un buen número, ya que, los diez primeros atributos de una columna y de la otra son los mismos, solo que cambiados de orden. Estos atributos serán:

1. *NuclearPVP*
2. *Month*
3. *TemperaturaGibraltar*
4. *TemperaturaMadrid*
5. *Hour*
6. *TemperaturaVigo*
7. *InterconexionesPVP*
8. *TermosolarPVP*
9. *FotovoltaicaPVP*
10. *DemandaPVP*

A continuación, se utilizará un método que filtre atributos y proporcione un conjunto que funcione bien como tal. Se escogerá una opción que no requiera de clasificador base. Para ello, puede seleccionarse el método de evaluación *CfsSubsetEval* y el método de búsqueda *GreedyStepwise*. Al utilizar *Use full training set* y *Cross-validation* con cuatro divisiones, los resultados son los siguientes (*figura 2.23*):

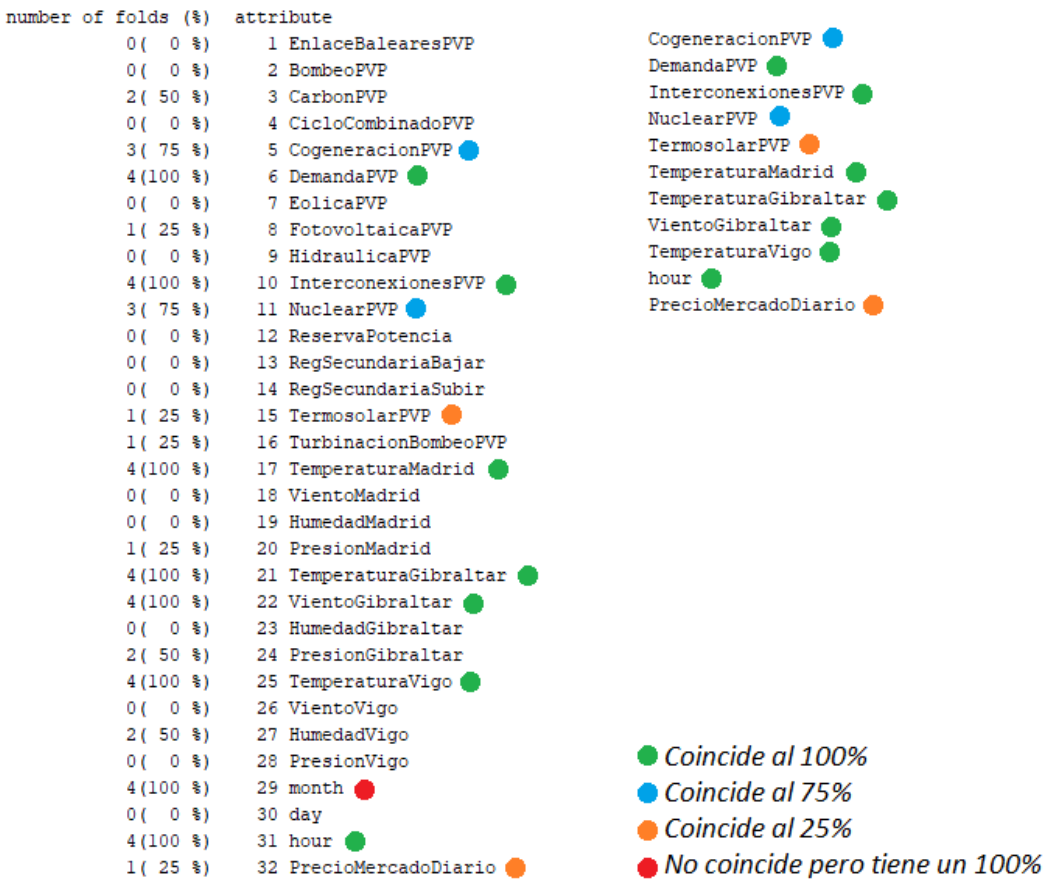


Figura 2.23 - Resultados con CfsSubsetEval

En la columna de la izquierda, se representan los resultados de la validación cruzada, marcados por un porcentaje. Este porcentaje será el número de veces que el atributo fue seleccionado de entre todas las divisiones. En la columna de la derecha se tienen los resultados de haber utilizado la muestra completa sin divisiones. En la leyenda se ha marcado con colores en que porcentaje coinciden los atributos seleccionados en una columna y otra. Hay atributos como *DemandaPVP* que aparece seleccionado a la derecha y también con el 100% a la izquierda. Otros como *TermosolarPVP*, aparece en la selección donde se usó toda la muestra, pero solo se escogió en una de las cuatro divisiones al utilizar la validación cruzada. También resulta curioso que el atributo *month* haya sido resultado elegido en las cuatro divisiones pero no al haber utilizado la muestra al completo. Si se utiliza la columna de la izquierda como referencia y se compara con la obtenida con el método *ranker*, se ve que la mayoría de atributos que se seleccionaron por su potencial de manera individual, aparecen en la nueva selección. Solo *TermosolarPVP* y *FotovoltaicaPVP* parece que no acaban de funcionar del todo bien en conjunto con otros. Al contrario sucede con *VientoGibraltar* que no aparece seleccionado por sus propiedades individuales, pero sí parece que trabaja bien en conjunto, siendo seleccionado en las cuatro divisones.

Por último, se probará el método *Wrapper*, que proporciona un conjunto de atributos que funcionan bien en conjunto pero para ello utiliza un clasificador base. Se probará tanto para árboles de decisión como para redes neuronales. Para ello se selecciona como método de

evaluación *ClassifierSubsetEval* y automáticamente, se tuviese el método de búsqueda que se tuviese, el programa pondrá un aviso de que se cambiará automáticamente a *GreedyStepwise*. Ahora habrá que acceder a la configuración del método de evaluación y seleccionar el clasificador, ya que por defecto aparecerá *ZeroR*, el cual solo discriminaba los datos según la clase mayoritaria. Se selecciona *J48 o* MultilayerPerceptron y *Cross-validation* con 4 divisiones. No se realizará el estudio sin divisiones de la muestra porque ya se prevé que el resultado debe ser menos fiable que si se divide en varias partes. Para el árbol de decisión, se deja la configuración de sus parámetros tal como aparece ya que se vio que no merecía la pena cambiar el *ConfidenceFactor*. Para la red neuronal se fija el *learningRate* en 0.6 y el *momentum* en 0.3, ya que fueron los parámetros que ofrecían mejores resultados. Para la selección de atributos con *J48* se requiere algo más de tiempo de computación que para realizar una clasificación como las que ya se hicieron, pero aun así, la espera es de pocos minutos. Para la red neuronal, si antes se requerían tiempos de varios minutos, ahora es posible tener que esperar una hora para que el programa arroje unos resultados. Todo esto teniendo en cuenta además, que el número de divisiones se ha reducido a 4 en lugar de 10. En la siguiente imagen (*figura 2.24*) pueden observarse ambos resultados.

Árbol de decisión

number of folds (%)	attribute
1(25 %)	1 EnlaceBalearesPVP
1(25 %)	2 BombeoPVP
1(25 %)	3 CarbonPVP
4(100 %)	4 CicloCombinadoPVP
0(0 %)	5 CogeneracionPVP
2(50 %)	6 DemandaPVP
2(50 %)	7 EolicaPVP
1(25 %)	8 FotovoltaicaPVP
2(50 %)	9 HidraulicaPVP
1(25 %)	10 InterconexionesPVP
3(75 %)	11 NuclearPVP
1(25 %)	12 ReservaPotencia
1(25 %)	13 RegSecundariaBajar
1(25 %)	14 RegSecundariaSubir
3(75 %)	15 TermosolarPVP
0(0 %)	16 TurbinacionBombeoPVP
1(25 %)	17 TemperaturaMadrid
3(75 %)	18 VientoMadrid
1(25 %)	19 HumedadMadrid
2(50 %)	20 PresionMadrid
0(0 %)	21 TemperaturaGibraltar
0(0 %)	22 VientoGibraltar
1(25 %)	23 HumedadGibraltar
2(50 %)	24 PresionGibraltar
0(0 %)	25 TemperaturaVigo
1(25 %)	26 VientoVigo
2(50 %)	27 HumedadVigo
0(0 %)	28 PresionVigo
3(75 %)	29 month
2(50 %)	30 day
1(25 %)	31 hour
2(50 %)	32 PrecioMercadoDiario

Red neuronal

number of folds (%)	attribute
3(75 %)	1 EnlaceBalearesPVP
1(25 %)	2 BombeoPVP
4(100 %)	3 CarbonPVP
1(25 %)	4 CicloCombinadoPVP
2(50 %)	5 CogeneracionPVP
3(75 %)	6 DemandaPVP
3(75 %)	7 EolicaPVP
1(25 %)	8 FotovoltaicaPVP
1(25 %)	9 HidraulicaPVP
3(75 %)	10 InterconexionesPVP
4(100 %)	11 NuclearPVP
0(0 %)	12 ReservaPotencia
1(25 %)	13 RegSecundariaBajar
1(25 %)	14 RegSecundariaSubir
2(50 %)	15 TermosolarPVP
1(25 %)	16 TurbinacionBombeoPVP
4(100 %)	17 TemperaturaMadrid
0(0 %)	18 VientoMadrid
4(100 %)	19 HumedadMadrid
3(75 %)	20 PresionMadrid
3(75 %)	21 TemperaturaGibraltar
2(50 %)	22 VientoGibraltar
3(75 %)	23 HumedadGibraltar
1(25 %)	24 PresionGibraltar
3(75 %)	25 TemperaturaVigo
0(0 %)	26 VientoVigo
3(75 %)	27 HumedadVigo
0(0 %)	28 PresionVigo
2(50 %)	29 month
2(50 %)	30 day
4(100 %)	31 hour
3(75 %)	32 PrecioMercadoDiario

Figura 2.24 - Resultado con método Wrapper

Como puede comprobarse, ahora ya empiezan a ver grandes diferencias de resultados con los de los métodos anteriores. Para el árbol de decisión, se escogerán los atributos que hayan sido seleccionados al menos el 50% de las ocasiones y para la red neuronal, los que hayan sido escogidos entre tres y cuatro veces. Esta distinción se hace porque para el árbol de decisión, hay pocos atributos que hayan sido seleccionados entre tres y cuatro veces. En cambio, para la red neuronal se cuenta ya con bastantes atributos en este rango. Para *J48* los atributos seleccionados serán:

1. *CicloCombinadoPVP*
2. *DemandaPVP*
3. *EolicaPVP*
4. *HidraulicaPVP*
5. *NuclearPVP*
6. *TermosolarPVP*
7. *VientoMadrid*
8. *PresionMadrid*
9. *PresionGibraltar*
10. *HumedadVigo*
11. *month*
12. *day*
13. *PrecioMercadoDiario*

Para la red neuronal serán:

1. EnlaceBalearesPVP
2. CarbonPVP
3. DemandaPVP
4. EolicaPVP
5. InterconexionPVP
6. NuclearPVP
7. TemperaturaMadrid
8. HumedadMadrid
9. PresionMadrid
10. TemperaturaGibraltar
11. HumedadGibraltar
12. TemperaturaVigo
13. HumedadVigo
14. Hour
15. PrecioMercadoDiario

Con esto, ya se ha obtenido una selección de atributos para cada uno de los métodos propuestos y para ambos clasificadores en el caso del método *Wrapper*. En la siguiente tabla (*tabla 2.*1) se puede ver a modo de resumen el resultado para cada uno de los métodos:

	Ranker	***GreedyStepwise***	***Wrapper (J48)***	***Wrapper (MultilayerPerceptron)***
1	*month*	*CogeneracionPVP*	*CicloCombinadoPVP*	*EnlaceBalearesPVP*
2	*TemperaturaMadrid*	*DemandaPVP*	*DemandaPVP*	*CarbonPVP*
3	*hour*	*InterconexionesPVP*	*EolicaPVP*	*DemandaPVP*
4	*TemperaturaGibraltar*	*NuclearPVP*	*HidraulicaPVP*	*EolicaPVP*
5	*TemperaturaVigo*	*TemperaturaMadrid*	*NuclearPVP*	*IntercionexionesPVP*
6	*DemandaPVP*	*TemperaturaGibraltar*	*TermosolarPVP*	*NuclearPVP*
7	*NuclearPVP*	*VientoGibraltar*	*ViendoMadrid*	*TemperaturaMadrid*
8	*FotovoltaicaPVP*	*TemperaturaVigo*	*PresionMadrid*	*HumedadMadrid*
9	*TermosolarPVP*	*month*	*PresionGibraltar*	*PresionMadrid*
10	*InterconexionesPVP*	*hour*	*HumedadVigo*	*TemperaturaGibraltar*
11	-	-	*month*	*HumedadGibraltar*
12	-	-	*day*	*TemperaturaVigo*
13	-	-	*PrecioMercadoDiario*	*HumedadVigo*
14	-	-	-	*hour*
15	-	-	-	*PrecioMercadoDiario*

Tabla 2.1 - Resultados selección de atributos

Por último, una vez obtenidos los distintos conjuntos de atributos mediante los distintos métodos de selección que se han explicado, lo que queda será evaluar si estos nuevos conjuntos, más reducidos, son más o menos efectivos respecto de usar la muestra completa. Para ello, se tendrán que eliminar los atributos que correspondan según el método de selección que se quiera evaluar. Esto es posible hacerlo de dos modos distintos. El primero es eliminar directamente los atributos desde la ventana de *Preprocess*. La segunda y más interesante, es utilizar un meta-clasificador que realice la selección de atributos y posteriormente haga la clasificación eliminando los atributos que no interesen. Para realizar la meta-clasificación habrá que ir a la ventana de *Classify* y seleccionar *AttributeSelectedClassifier* de la lista de clasificadores. Desde la ventana de configuración de este clasificador, habrá que elegir el método de evaluación y el método de búsqueda del mismo modo que se hizo anteriormente. Desde esta ventana se puede acceder a las propias configuraciones de cada método de búsqueda o evaluación tal como se hacía en la ventana de *Select attributes*. También se debe elegir un clasificador, para que una vez se haya realizado la selección, se realice seguidamente la clasificación con los atributos seleccionados. Lo único que hay que tener en cuenta, es que si se escoge un método *Ranker*, este no desechará atributos a la hora de hacer la posterior clasificación siempre que no se le diga lo contrario. Para cambiarlo, se accede a la configuración de *Ranker* y en el campo *numToSelect*, se debe cambiar el valor '*-1*' por el número de atributos que se quieran escoger. Así, se realizará la clasificación con los '*n*' atributos mejor evaluados. Esto es útil si no se conocen los atributos que saldrían seleccionados, pero como esta parte ya se ha llevado a cabo, no se perderá tiempo en esperar unos resultados que ya se obtuvieron.

Por lo tanto, se seleccionarán directamente en *Preprocess* los atributos de la tabla anterior (*tabla 2.1*) y se aplicará el mismo estudio de clasificación que ya se hizo con la muestra completa.

2.4.3.1. MÉTODO *RANKER*

Lo primero, como ya se ha dicho, será seleccionar los atributos correspondientes. En este caso se seleccionan los 10 mejores que se obtuvieron con el método *Ranker*. Posteriormente se abre la ventana de *Classify* y se selecciona el algoritmo *J48*. En la siguiente tabla (*tabla 2.2*) se muestran los resultados generales obtenidos con la muestra completa y con la muestra reducida.

	Muestra completa	Muestra dada por *Ranker*
Use full training set	96%	90.5%
Cross-validation (10 divisiones)	74.5%	71.3%
Cross-validation (5 divisiones)	75%	71.7%
Cross-validation (20 divisiones)	75.6%	70.9%
Percentage Split (66%)	72.3%	66.4%
Percentage Split (99%)	84%	76%
Percentage Split (99%) (muestra ordenada)	76%	68%

Tabla 2.2 - Resultados ranker y J48

En principio, todos los resultados empeoran bastante al eliminar los atributos que tuvieron peor evaluación. Si al último de los resultados, el del 99% con la muestra ordenada, se le reduce el *ConfidenceFactor* a un valor cercano a cero (0.01), el porcentaje de aciertos sube a 76%. Si además se observa la matriz de confusión (*Ecuación 2.)*, se ve que se clasifica erróneamente como desvíos a subir, una vez, por cada cinco veces que se clasifica erróneamente como desvío a bajar.

$$\begin{pmatrix} 11 & 5 \\ 1 & 8 \end{pmatrix} \qquad (\textit{Ecuación 2.})$$

Por lo tanto se va a utilizar el meta-clasificador *CostSensitiveClassifier*, el cual ya se explicó previamente. Se selecciona *J48* con el *confidenceFactor* en 0.01 y en la matriz de costes, se

multiplica por dos el coste de clasificar como desvío a bajar. Ahora el porcentaje de acierto sube a 80%, lo que son 20 aciertos de 25 horas predichas. Anteriormente, en el estudio con la muestra completa, no se redujo el *confidenceFactor* a un valor tan bajo, pero viendo que con menos atributos funciona bien, se prueba también con la muestra completa. Se hace para el mismo caso de 99% y la muestra ordenada y se obtiene un 84% de aciertos. Esto se consigue sin haber tenido que modificar la matriz de costes, pero resulta una matriz de confusión simétrica, por lo que no será posible conseguir una mejora con este método. En definitiva, por el momento no parece que el método *Ranker* de buenos resultados para árboles de decisión, además hay que tener en cuenta que es un algoritmo tan rápido, que no se consigue una mejora sustancial en el tiempo de computación.

En el caso de la red neuronal, en general se obtienen peores resultados al reducir el número de atributos al conjunto dado. Pero en concreto, y en el caso que más interesa, al dividir la muestra en 99% para entrenamiento y 1% de testeo, se obtiene un 88% de aciertos. Esto se ha hecho con la muestra ordenada y se llegan a acertar 22 horas de 25. Con la muestra completa y ordenada, el porcentaje de aciertos se veía muy reducido y había que modificar los parámetros de la red neuronal para hacerla más compleja. Ahora, se prueba tanto hacerla más compleja como hacerla más sencilla y en todos los casos se obtienen peores resultados. Con ello se puede entender que para la cantidad anterior de datos se necesitaba una red más compleja, pero que para la cantidad actual, los valores que vienen por defecto, definen la red ideal para este conjunto de datos. Antes se llegó a conseguir un 92% de aciertos pero también hay que tener en cuenta que el tiempo de computación ahora se ha podido ver reducido a la mitad.

2.4.3.2. MÉTODO GREEDYSTEPWISE

Hay que recordar, que para el método *ranker* se evaluaban por separado los atributos y se hacía una selección de los 10 mejores. Para el método *GreedyStepWise*, no se evalúa el valor de cada atributo si no que se proporciona un conjunto que debe de dar buenos resultados por sí mismo.

Haciendo la evaluación del método para el árbol de decisión (*J48*), se vuelven a ver de manera general peores resultados, excepto cuando se llega a la división con orden del 99%. Si se fija el *confidenceFactor* a 0.01, se obtiene un 80% de acierto. La matriz de confusión es la siguiente (*ecuación 3.*):

$$\begin{pmatrix} 13 & 3 \\ 2 & 7 \end{pmatrix} \qquad (\textit{Ecuación 3.})$$

Se ve que se cometen tres errores clasificando como desvío a bajar por cada dos errores de clasificación como desvío a subir. Se prueba a aumentar el coste de clasificar como desvío a bajar, de 1 a 1.1. Hecho esto, el porcentaje de aciertos sube a 84%, lo que suponen 21 aciertos de 25. La matriz de confusión resultante es ahora simétrica, por lo que ya no tendrá sentido

seguir modificando los costes. Con esto, para árboles de decisión se han conseguido los mismos resultados bajo el mismo procedimiento, utilizando tanto un método como el otro.

Para la red neuronal, se vuelve a comprobar como antes, que los resultados en general son bastante malos. Incluso la división con orden del 99% que se hizo con el método *ranker*, la cual dio buenos resultados, esta vez da un 36% de acierto. Antes parecía que modificando los parámetros del algoritmo no se conseguían mejores resultados, y ahora sigue siendo así, excepto cuando se le reduce el *momentum*. Reduciendo este parámetro a 0.1, dejando el resto de parámetros por defecto, se consigue un acierto del 88%, es decir, 22 aciertos de 25 horas predichas. Resulta curioso que de una muestra a otra (de *Ranker* a *GreedyStepWise*), solo se hayan cambiado los atributos *FotovoltaicaPVP* y *TermosolarPVP* por *CogeneraciónPVP* y *VientoGibraltar*, y ya se haya generado tanta diferencia de resultados, al menos, de forma aparente, ya que modificando ligeramente uno de los parámetros se ha llegado al mismo resultado.

2.4.3.3. MÉTODO *WRAPPER*

Ahora se probará para ambos clasificadores, las selecciones de atributos realizadas por ambos algoritmos. Así se verá cuanto de bueno es un clasificador seleccionando atributos tanto para sí mismo como para el otro clasificador.

Se comenzará por los atributos seleccionados mediante el clasificador *J48* y a estos se les aplicará la clasificación según el mismo algoritmo. Por lo tanto, una vez filtrados los atributos en *Preprocess*, se abre la ventana de *Classify* y se escoge el *J48*. Lo primero que se nota, es que para la validación cruzada con 10 divisiones, se mejora el resultado ligeramente, de haberlo hecho con la muestra de atributos completa a la filtrada. Antes se obtenía 74.5% de aciertos y ahora se llega a 76.5%. Incluso se nota que si se van aumentando el número de divisiones, el porcentaje de aciertos sigue aumentando pero muy poco a poco. El máximo número de divisiones con el que se ha simulado para el caso actual es de 100 y dando un 78% de aciertos. Luego, si se realiza la división con 66% para entrenamiento y el resto para testeo, se obtiene 76%, siendo 72.3% el que se obtuvo utilizando todos los atributos. Directamente se pasará a hacer la división de 99% con los datos ordenados. Se obtiene un 88% de aciertos, 22 de 25 horas acertadas. Es importante resaltar que no se ha necesitado modificar parámetros ni la matriz de coste para obtener este resultado. También hay que decir que los tiempos de computación para obtener la clasificación de atributos y generar el actual modelo, son mínimos. Por ello, para clasificar con árboles de decisión, parece ser conveniente realizar primero una selección de atributos con un método *wrapper* que lleve implícito el uso del algoritmo *J48*. Aun quedará probar si la selección que se hizo utilizando redes neuronales, tiene mejor o peor efecto sobre la clasificación que hacen los árboles de decisión.

Utilizando la selección de atributos anterior para realizar una clasificación utilizando el *MultilayerPerceptron*, en general da peores resultados. No llegando ni al 75% utilizando la misma muestra tanto para entrenar como para testear. Si se pasa directamente a realizar la división del 99% y a testear el 1% restante, se obtiene un 80% de aciertos, es decir, 20 aciertos

de 25 horas. Se prueba a cambiar parámetros, tanto para hacer la red más compleja como para hacerla más simple y en ambos casos el porcentaje de aciertos cae bastante bajo. Por lo tanto parece que los parámetros por defecto son adecuados. Finalmente, se puede decir que no es mal resultado obtener un 80%, pero que utilizando el árbol de decisión se conseguía un mejor resultado y con un tiempo de computación casi instantáneo. Es lógico, ya que los atributos fueron seleccionados mediante un árbol de decisión y no mediante una red neuronal.

Ahora se invertirá el orden anterior y se probará la última selección de atributos, primero con la clasificación que hace la red neuronal y después con el árbol de decisión. Esto se hará porque esta selección de atributos se realizó mediante un *wrapper* con red neuronal y parece el orden lógico de seguir.

Lo primero es actualizar la lista de atributos seleccionados para realizar la clasificación. Hasta ahora, la mayor selección de atributos se hizo con el *J48* que seleccionó 13 atributos. La actual selección tiene 15 atributos y es la mayor de las cuatro que se han obtenido. Después se vuelve a la ventana de clasificación y se selecciona la red neuronal para empezar a clasificar. En principio los resultados para el testeo con la muestra completa y la validación cruzada no son especialmente buenas. Se obtuvieron mejores resultados con la muestra de atributos completa. Al realizar la simulación para la división de 99% con orden, se obtiene un 84% de aciertos, o 21 horas de 25 acertadas. No es un mal resultado, pero se podría esperar algo más de la red neuronal, teniendo en cuenta que los atributos fueron seleccionados mediante el mismo método y que requiere de más tiempo de computación que el *J48*. Además, se prueba a modificar los parámetros de la red neuronal y el resultado lo único que hace es empeorar.

Por último, se utilizará esta última selección de atributos para clasificar los datos con el árbol de decisión. En principio, se realizan el testeo con el modelo generado sobre la misma muestra y también la simulación con 10 divisiones. Los resultados son ligeramente peores que los resultados de haber utilizado el *J48,* pero no se alejan notablemente de ellos. En cambio, cuando se divide la muestra al 99% con orden, el porcentaje de aciertos cae fuertemente hasta el 60%. Se intentan modificar parámetros de entrada al algoritmo pero no se consigue mejorar el resultado.

Con esto, se dan por probados de manera general, los métodos de selección de atributos. Entonces, según lo experimentado, es posible decir que a los árboles de decisión les viene bien acotar los atributos que se utilicen, aunque conviene utilizar el propio algoritmo con el método *wrapper* para obtener esta selección. Es un algoritmo bastante rápido generando modelos y seleccionando atributos, con lo que merece la pena realizar suficientes pruebas de parámetros y atributos hasta sacar buenos resultados. Esto es posible conseguirlo, ya que bajo determinadas condiciones, se han obtenido resultados muy cercanos al 90% de aciertos.

Por otro lado, las redes neuronales, con un mayor tiempo de computación, parece que son capaces de generar modelos más complejos y más efectivos cuando se tratan grandes cantidades de datos. Se podía esperar un mejor comportamiento del algoritmo utilizando su propia clasificación de atributos pero aun así no deja de ser un clasificador malo bajo estas condiciones. De cualquier modo, no se llega a justificar el tiempo que se tarda en obtener una selección de atributos con la red neuronal y el porcentaje de aciertos que se obtiene finalmente. También hay que tener en cuenta que quizás, la selección de los datos iniciales del estudio no

es la más correcta, ya que, se recuerda que la muestra, contenía los datos para todos los fines de semana del año. Esto puede ser que de tan buenos resultados como malos. Puede parecer lógico comparar horas que cumplan esta condición de fin de semana de 2017, pero también se puede pensar que un fin de semana de Enero no tenga nada que ver con uno de Julio. No se puede saber en principio y es necesario realizar distintas pruebas. Aun así, se ha visto que es posible obtener resultados muy positivos con ambos métodos, teniendo las consideraciones pertinentes.

2.4.4. VISUALIZACIÓN DE ATRIBUTOS

Otra manera de realizar una preselección de atributos de manera sencilla es la pre-visualización gráfica. En cualquier momento, una vez cargados los datos de los atributos en el programa, es posible visualizar gráficamente cómo funcionan los atributos por parejas. Si se abre la ventana de *Visualize* (*figura 2.25*), se puede ver una cuadrícula donde en cada una de las celdas se encuentra una gráfica con la distribución de datos de cada atributo, indicando con colores (rojo y azul) si corresponden a desvíos a bajar o a subir.

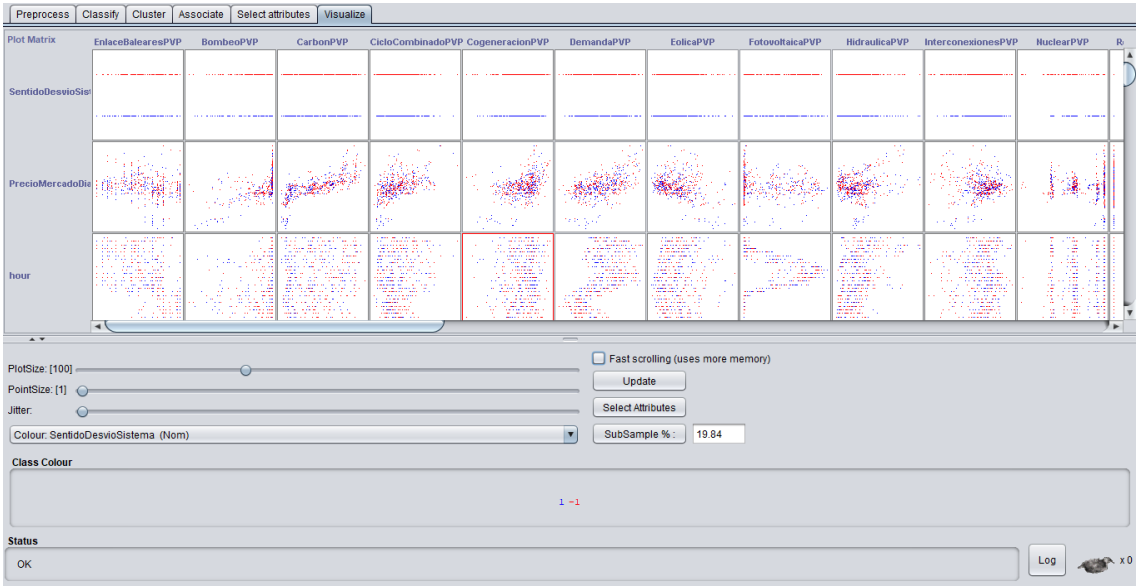


Figura 2.25 - Ventana de Visualize

Por ejemplo, si se observan las selecciones de atributos que se han realizado, se puede ver que tanto el atributo *DemandaPVP* como el de *NuclearPVP*, aparecen en las cuatro selecciones realizadas. Por ello, se entiende que son buenos atributos y puede ser interesante comparar uno con otro en la ventana de visualización (*figura 2.26*).

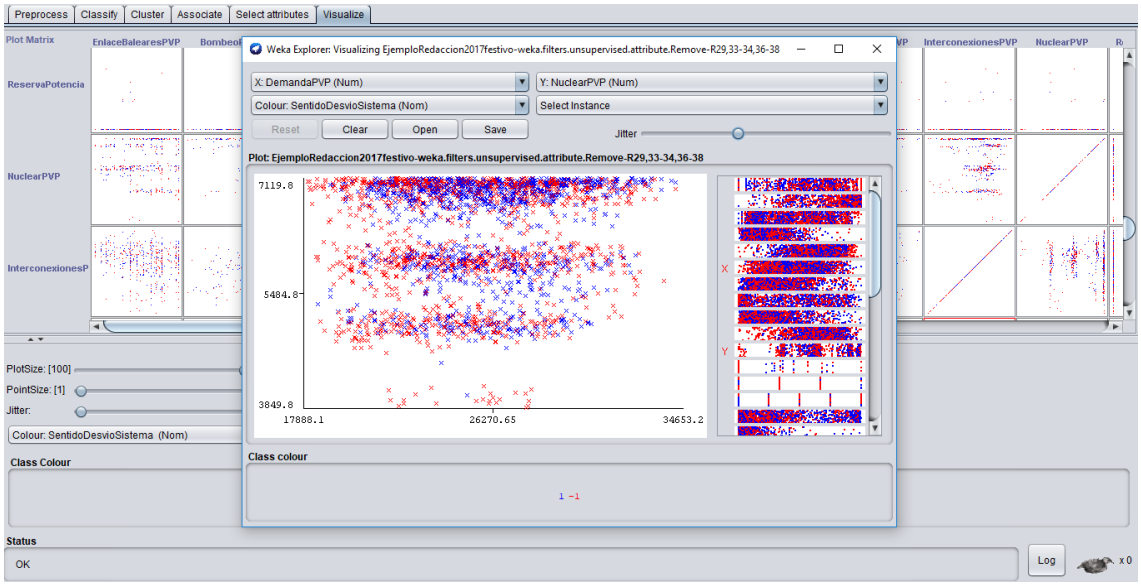


Figura 2.26 - Visualización Demanda-Nuclear

En el eje de abscisas se tienen los valores de la demanda y en el eje de ordenadas, los del resultado de la nuclear. Los distintos puntos marcados son los distintos valores que se han dado en cada hora y el color azul o rojo, indica el sentido del desvío para ese momento. Realmente, es una clasificación tan compleja y se requiere de la información de tantas variables, que es complicado ver una clara diferencia entre si un atributo es bueno o no mediante este método. En la siguiente imagen (*figura 2.27*), se observa un ejemplo de atributos que realizan la clasificación de manera ideal. Para el caso hipotético representado se clasifican según tres clases en lugar de dos, como en el actual estudio. De ahí que aparezcan tres colores en lugar de dos. De por sí, los dos atributos son capaces de separar perfectamente en las tres clases, como se representa con las líneas negras. Antes de entrar en cualquier estudio de detalle sobre cualquier muestra de atributos, es aconsejable dirigirse a esta ventana para ver si se puede obtener alguna pista adicional sobre cómo enfocar el estudio.

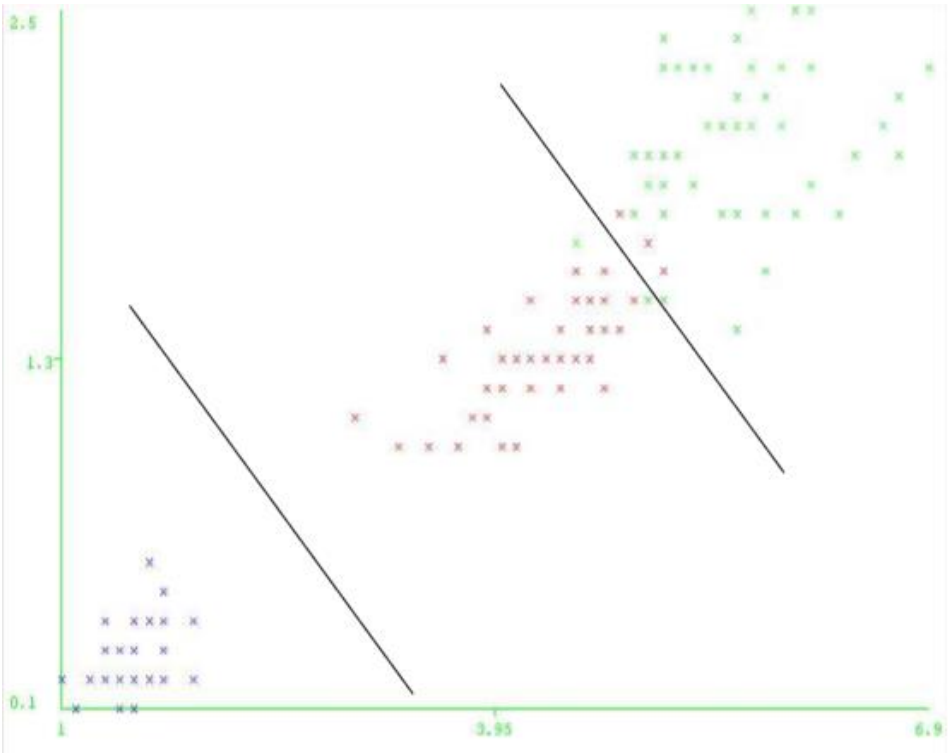


Figura 2.27 - Atributos ideales para clasificar

2.4.5. EL EXPERIMENTER

El *Experimenter* es la última de las herramientas que se explicarán y una de las más útiles del programa. No aporta nada nuevo en cuanto a métodos de predicción, ya que los algoritmos que aplica y los parámetros de entrada serán los mismos que ya se han explicado. Lo que la hace interesante, es la posibilidad de programar experimentos sobre distintos grupos de datos y utilizando distintos métodos de predicción, obteniendo una comparación final de los resultados obtenidos. Es decir, el *Experimenter* nos dirá si las diferencias aparentes en porcentajes de aciertos de distintos algoritmos son estadísticamente significativas o son debidas al azar. Para abrir esta herramienta, será necesario volver a la ventana de inicio previa al *Explorer*.

Una vez en la ventana inicial (*figura 2.28*), se identifican tres pestañas en la parte superior: *Setup, Run* y *Analyse*. En la primera de ellas se establecerán todas las condiciones, datos de entrada y algoritmos que se aplicarán durante el experimento. En la segunda, se dará inicio al experimento y se verá la evolución del mismo. Esto es importante, ya que se si van a realizar múltiples simulaciones sobre distintos grupos de datos, el tiempo de computación puede ser significativo y desde esta ventana, se podrá ver qué conjunto de datos y qué algoritmos se están simulando en cada momento. En la última, se encuentra el visor de los resultados y también será posible elegir qué información se presenta por pantalla y el formato de los datos para su exportación.

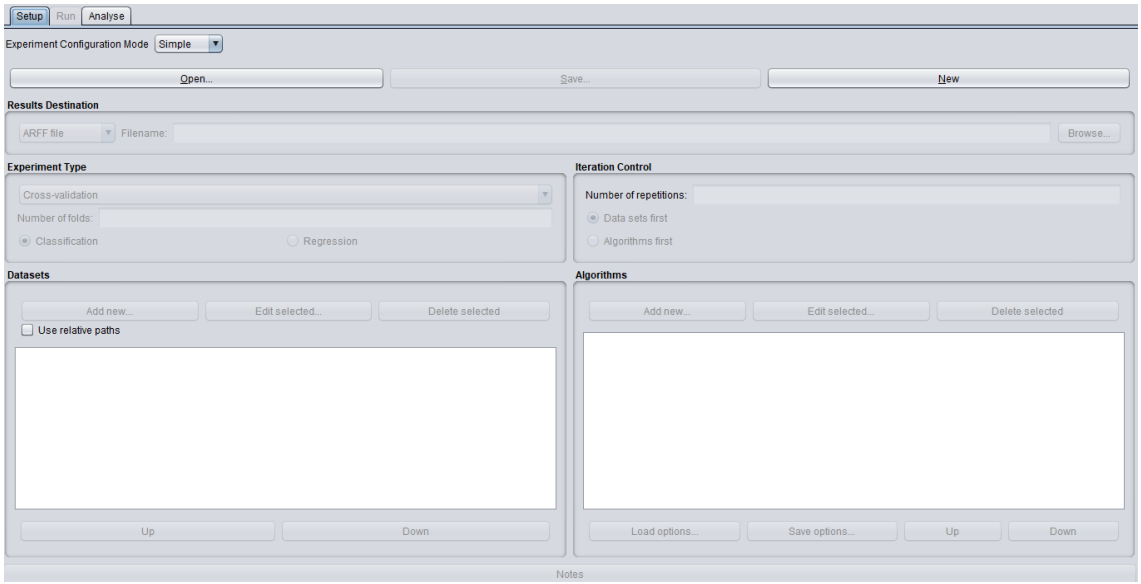


Figura 2.28 - Ventana del Experimenter

Lo primero que se debe hacer, es establecer las condiciones del experimento en *Setup*. La primera opción que aparece en la pantalla, es la de escoger entre realizar una configuración simple o avanzada del experimento (*Experiment Configuration Mode*). En principio se escogerá la configuración simple, la cual aparece por defecto y resulta suficiente para el estudio que se va a llevar a cabo. Seguidamente habrá que escoger entre realizar un nuevo experimento o abrir uno que se hubiese guardado previamente. Como es la primera vez que se utiliza esta herramienta, se seleccionará *New*. Hecho esto, se habrán habilitado todas las pestañas y ventanas de selección que antes aparecían en gris. Ahora habrá que crear el fichero al que se volcarán todos los resultados del experimento. Para ello, se hará click en *Browse*, se selecciona el lugar donde se quiere guardar el fichero y se le pone un nombre cualquiera. Con esto, aparecerá la ruta del fichero creado en la ventana de *Filename*. Este fichero de salida se le puede dar tres formatos distintos: ARFF, CSV o JDBC. En principio ARFF será un formato válido, ya que hasta ahora es el que se ha manejado para todos los datos de entrada y salida. Es el formato que aparece por defecto, pero aun así, una vez creado el fichero de salida, se debe volver a hacer click en *ARFF file* para que se le aplique al fichero. Cuando se hace esto, automáticamente aparece *.arff* en la ruta del fichero de salida. De no hacerlo, puede aparecer un error al querer comenzar la simulación más adelante.

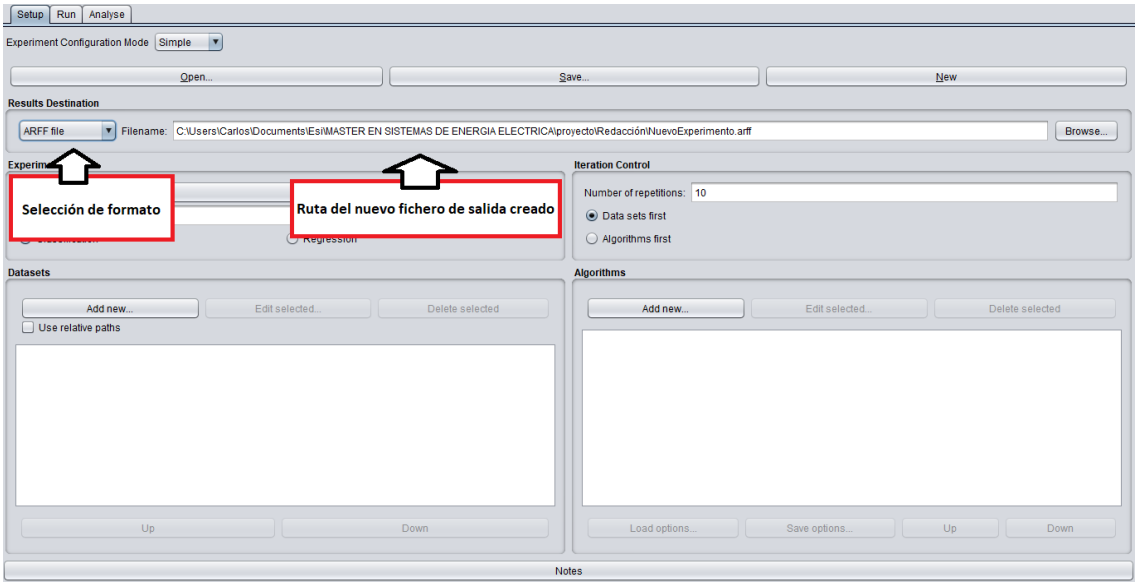


Figura 2.29 - Creación de fichero de salida del experimento

Ahora se debe elegir el tipo de experimento en la ventana de *Experiment Type*. Será posible escoger entre validación cruzada (con un número de divisiones a elegir) y división de la muestra en entrenamiento y testeo con un porcentaje fijado. Para esta última opción se puede elegir tanto manteniendo el orden como sin él. También se puede escoger entre realizar una clasificación o una regresión. Para el experimento actual se escogerá la división del 99% que ya se ha realizado previamente y manteniendo el orden de los datos. Por supuesto también se escogerá clasificación. En la ventana de *Iteration Control* se tendrá habilitada para este caso dos opciones: *Data sets first* y *Algorithms first*. Si se selecciona la primera, se aplicarán todos los algoritmos a un mismo conjunto de datos antes de pasar al siguiente. Si se escoge la segunda, sucede lo contrario: para todos los conjuntos de datos se aplica un algoritmo y así sucesivamente hasta aplicarlos todos. En este caso se escogerá *Data sets first*.

Por último, se deben escoger los grupos de datos que se quieren clasificar (ventana de *Datasets*) y los algoritmos que se utilizarán para ello (ventana de *Algorithms*). Hasta ahora, solo se había simulado con un solo grupo de datos, pero como ahora se quiere ver la funcionalidad completa de esta herramienta, se añadirá un nuevo grupo. Antes se utilizaron los datos para los días festivos de 2017 y ahora se añadirá un segundo conjunto que pertenezca a los datos de los días festivos de 2016. Para los algoritmos, se seleccionarán distintos casos dentro de los ya simulados. El primero será el clasificador *ZeroR*, lo cual servirá para marcar la referencia que han de superar el resto. Para árboles de decisión se realizarán tres simulaciones utilizando el *J48.* Las dos primeras fijando el *confidenceFactor* en 0.25 y 0.01. En la tercera se hará una preselección de atributos con método *Wrapper* que utilice el *J48* como clasificador base, ya que fue la que mejores resultados proporcionó anteriormente. Para redes neuronales se puede realizar una primera clasificación utilizando el *MultilayerPerceptron* sin modificar sus parámetros, otra modificando ligeramente sus parámetros y una última utilizando una preselección de atributos con *GreedyStepwise*. Se utiliza este método y no otro debido a que se vio que no merecía la pena el tiempo de computación para realizar una clasificación más compleja.

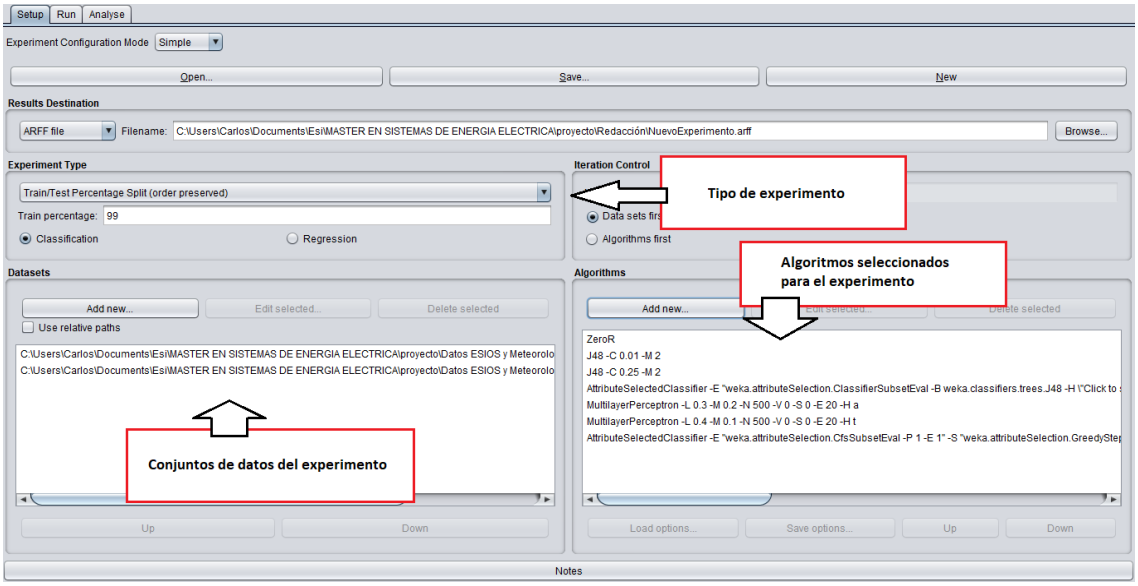


Figura 2.30 - Elección de conjuntos de datos y algoritmos para el experimento

Esta selección de algoritmos con distintos parámetros y configuraciones es posible guardarla si se quiere utilizar en futuros experimentos con otros conjuntos de datos. También es posible guardar la configuración completa que se ha hecho de todo el experimento. Una vez establecidas todas las condiciones, se selecciona la pestaña *Run* y se pulsa *Start*. En la ventana de *Log* aparecen los distintos hitos que se van cumpliendo a lo largo de la simulación. En caso de no existir ningún error en la configuración y en la entrada de datos, la secuencia de mensajes que aparecerán son los que se pueden ver en la siguiente imagen (*figura 2.31*). En la barra de *Status* se podrá ver a lo largo de la simulación, las distintas acciones que se van realizando, indicando conjunto de datos y algoritmo que se está utilizando en ese momento.

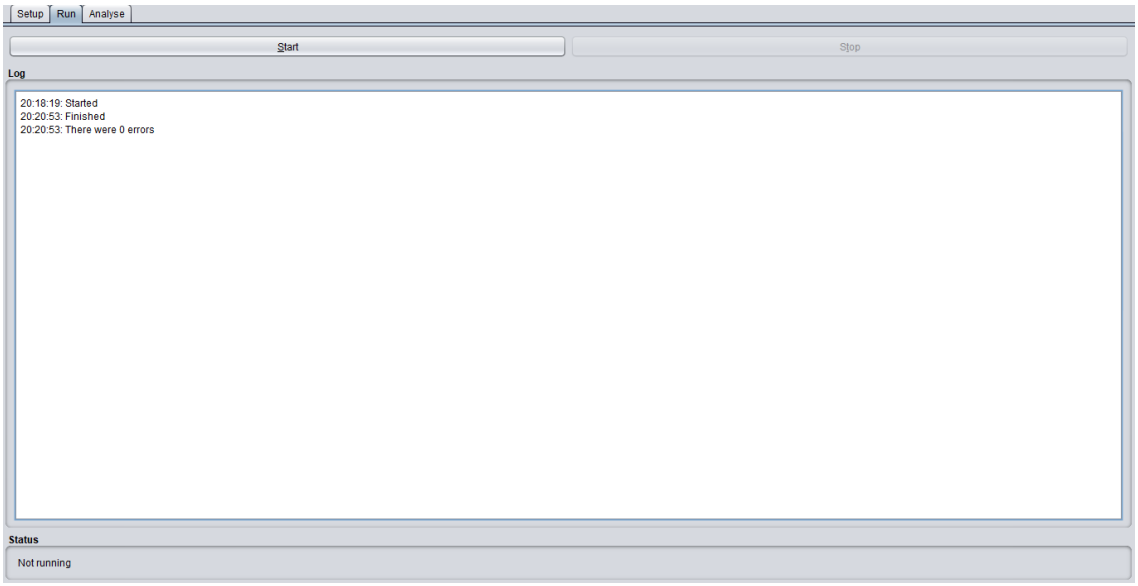


Figura 2.31 - Ventana del estado del experimento

Una vez el experimento ha acabado y aparece el mensaje de “cero errores”, hay que acceder a la pestaña de *Analyse*. En esta ventana es donde se realizará la visualización de los resultados del experimento. Lo primero será hacerle saber al programa, los resultados de qué experimento debe cargar. Si el experimento ha sido recién concluido, habrá que hacer click en *Experiment*. Como todo experimento vuelca sus resultados en un fichero al cual se le ha puesto nombre al principio, será posible también acceder a los resultados de cualquier experimento pasado desde la pestaña *File*. Para el caso actual, se pulsa la pestaña *Experiment* y automáticamente aparecen en la ventana de *Test output* los distintos experimentos que se han llevado a cabo. Para este caso, aparecen los siete casos distintos de algoritmos que se han aplicado. Si se pulsa *Swap* en la columna de la izquierda, aparecerán en su lugar los conjuntos de datos utilizados para la simulación, con lo que si no se conoce bien que contiene un fichero de experimento, de este modo es posible ver lo que contiene.

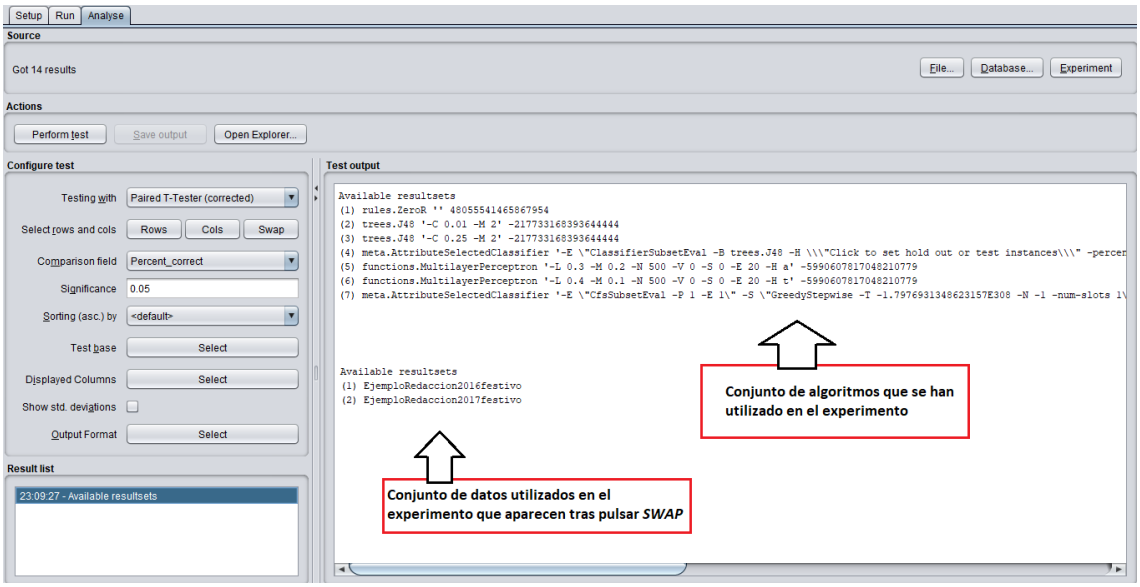


Figura 2.32 - Ventana de visualización de resultados del experimento

Pulsando en *Perform test* se devolverán por pantalla los resultados del experimento. Previamente se pueden modificar las opciones de visualización en la columna de la izquierda (*Configure test*). Si esto no se ha realizado para cuando se pulsa *Perform test* no hay problema, ya que se tratan únicamente de opciones de visualización y se puede volver a modificar la configuración y volver a pulsarlo para que aparezcan los resultados de la manera requerida. Lo primero que se hará antes de obtener la visualización, será abrir la pestaña de *Output Format*. Se elegirá la opción de *Plain Text*, lo cual es lo más recomendable para una primera visualización y se seleccionará la opción *Show Average* (muestra una media de efectividad para cada algoritmo utilizado a partir de los resultados individuales que cada algoritmo ha obtenido con cada grupo de datos). En *Advanced setup* existen más parámetros que se pueden modificar para presentar la información de un modo u otro, aunque esto resulta ya menos importante y dependerá de la elección del usuario. Antes de seguir comentando el resto de opciones de la

columna de configuración, resulta conveniente ver los resultados para ver mejor el sentido que tiene cambiar parámetros de visualización (*figura 2.33*).

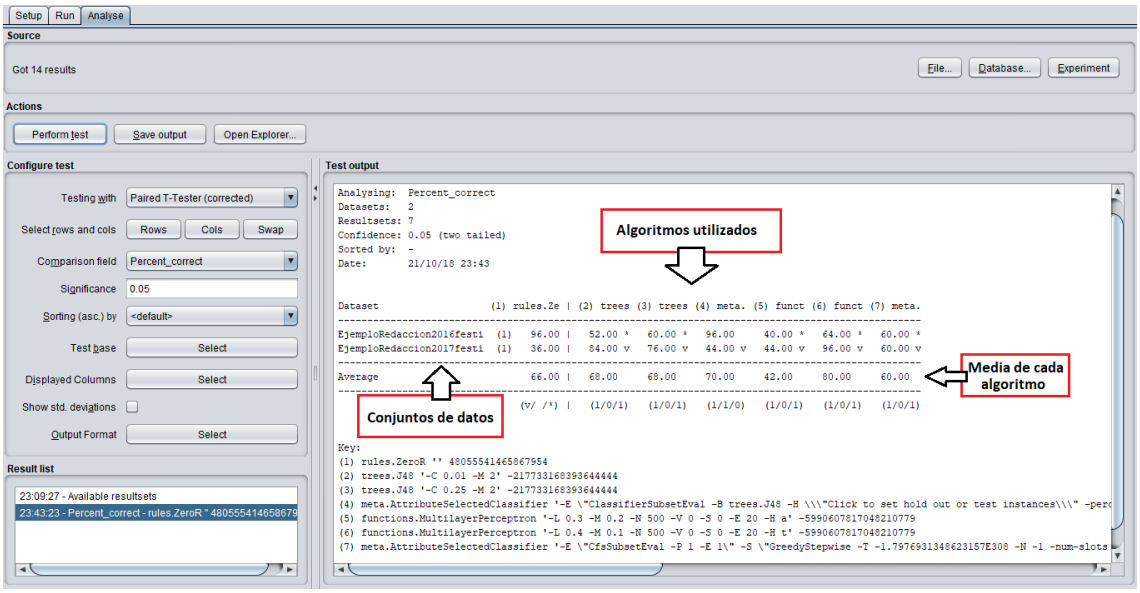


Figura 2.33 - Visualización de resultados del experimento

Una vez se obtienen los resultados en la ventana de visualización, se ve como aparecen distribuidos en una tabla con algoritmos en uno de los ejes y conjuntos de datos en el otro. En el caso mostrado en la imagen, se evalúan los distintos clasificadores según los aciertos que obtienen con unos conjuntos de datos y otros. Si se volviese a pulsar *Swap*, serían los conjuntos de datos los que serían evaluados para cada uno de los algoritmos. Es posible ver como todos los algoritmos muestran sus resultados y aparte son comparados con un algoritmo de base. Por defecto aparece *ZeroR* al haber sido elegido el primero. Esto se ha hecho a propósito ya que en teoría, este es el mínimo a acertar. Aun así, es posible cambiar el algoritmo de base en la pestaña *Test base*. La comparación se realiza con una nomenclatura concreta. Cada uno de los resultados que no correspondan al clasificador *ZeroR*, tendrán un asterisco (*), una "*v*" o no tendrán nada. En el caso del asterisco, significará que el clasificador obtiene un resultado significativamente peor que el clasificador de base. En el caso de la "*v*" significará que el resultado mejora en gran medida el resultado proporcionado por el clasificador base. En el caso de que no cuente con ningún símbolo significará que el resultado no presenta grandes diferencias con el del clasificador base. Así por ejemplo, en la anterior imagen (*figura 2.33*), parece que ninguno de los métodos mejora al clasificador *ZeroR* para los datos de 2016 de ese día concreto. Resulta curioso el dato, ya que con *ZeroR* se obtiene un 96% de aciertos en ese caso. Eso quiere decir que para ese día, el 96% de los desvíos fueron en el mismo sentido y que *ZeroR* tuvo la suerte de acertarlo. Eso no quiere decir que sea un buen clasificador, ya que del mismo modo, si se hubiera dado un día donde todos los desvíos fuesen al contrario, *ZeroR* hubiera obtenido un resultado muy cercano a 0%. Para los resultados de 2017 se ve como todos los clasificadores mejoran a *ZeroR* con mejor o peor resultado según qué casos. Es posible también obtener los datos de salida en otros formatos que no sea el porcentaje de aciertos. Se pueden ver también

como número de aciertos, número de fallos, tiempo que ha tardado cada algoritmo en obtener su modelo y una larga lista de opciones que permitirá ver los resultados desde distintos puntos de vista.

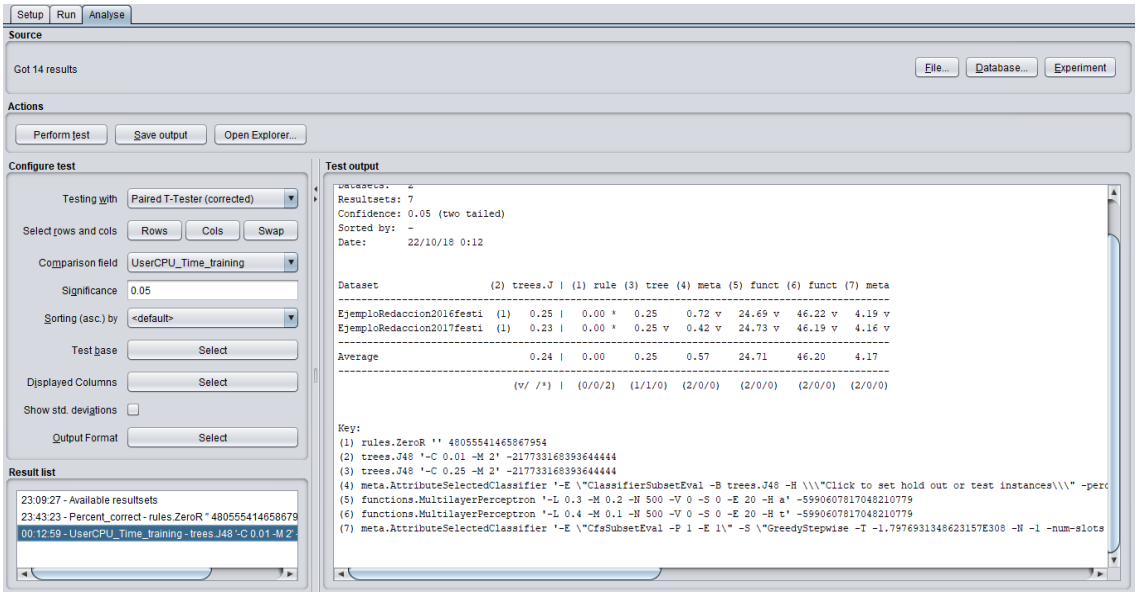


Figura 2.34 - Visualización de los tiempos de cada algoritmo en el experimento

Para la imagen anterior (*figura 2.34*), se pueden ver los distintos algoritmos clasificados según el tiempo de computación que ha utilizado cada uno para generar su modelo. Se ve como los clasificadores asociados a árboles de decisión son bastante rápidos en contraposición a la lentitud de las redes neuronales, que solo ven reducido los tiempos en el caso 7, donde se realizó una selección previa de atributos.

En definitiva, son muchas las opciones a elegir y es numerosa la cantidad de información que se puede obtener con esta herramienta. Corresponderá a la propia experiencia de cada usuario el explorar las distintas opciones y posibilidades que aquí se ofrecen. Un último punto importante a añadir, es que todos estos resultados son exportables en distintos formatos. Concretamente, al exportarlo en formato HTML, se podrán abrir en EXCEL en forma de columnas ordenadas. Esto resultará útil si se manejan varios clasificadores diferentes con numerosos conjuntos de datos distintos. Una vez se tengan la configuración de la visualización y del formato del modo requerido, se hará click en *Save output* y se extraerá el experimento para poder ser tratado desde otro programa si se requiere.

Por último, se aclara que este apartado tiene únicamente la intención de demostrar y enseñar las capacidades y herramientas del programa. Los resultados mostrados se han obtenido utilizando datos reales y por lo tanto arrojan resultados reales, pero no deberán ser tenidos en cuenta a nivel de conclusiones generales, ya que son fruto de haber escogido unas condiciones muy concretas y una ventana temporal de la muestra específica. De haber cambiado las condiciones o de haber simulado con muestras distintas se podría haber llegado a conclusiones distintas.

3. HIPÓTESIS Y RESULTADOS DEL PRIMER ESTUDIO.

Hasta este momento, se ha mostrado la información con la que se va a trabajar y se han visto las distintas funciones y posibilidades del programa haciendo un recorrido general por el mismo utilizando una muestra de los datos descargados. También se ha establecido el objetivo de la predicción, el cual es el de generar un modelo que sea posible aplicar a la información que se tenga antes de las 17:00 horas para predecir lo que ocurra en las 24 horas del día siguiente. Una vez llegado a este punto, lo siguiente será comenzar a realizar pruebas con la información de los cuatro años de partida para tratar de encontrar una manera de conseguir buenas predicciones con ella.

Ha medida que se han ido descargando y añadiendo los datos al fichero de partida se han realizado pruebas con los distintos algoritmos y las distintas opciones de entrenamiento que existen. Una de las primeras impresiones que se tienen, es que mientras más atributos se le añadan al fichero de entrada, mejores porcentajes de aciertos se van obteniendo. Si se parte de la idea de que todos los atributos que se añaden están de alguna manera relacionados con la ocurrencia de posibles desvíos, resulta lógico que mientras más se añadan, más información de entrenamiento se está aportando y más posibilidades de que al testear, se encuentren casos parecidos a los de entrenamiento. Pero aun así, llega un momento en el que seguir añadiendo atributos no aporta mejoras significativas en la predicción y también se corre el riesgo de que alguno de ellos genere falseamiento de la predicción. Por otro lado, también se detecta que al agrupar los datos en distintos conjuntos de características similares se obtienen mejores resultados. Eso sí, estos conjuntos deben tener unas dimensiones mínimas o de lo contrario no se realizan buenas clasificaciones. Las distintas posibilidades de agrupación de datos son numerosas, por lo que dependerá de cada usuario el realizar sus propias hipótesis a la hora dividir la muestra total de los cuatro años en distintos conjuntos.

Tras haber realizado algunas pruebas preliminares con el programa y haber obtenido las conclusiones que se acaban de comentar, se decide dividir la muestra discriminado por estaciones y por días laborales (de lunes a viernes) y festivos (sábados y domingos). Para las estaciones, se ha considerado:

- Invierno: Enero, Febrero y Marzo.
- Primavera: Abril, Mayo y Junio.
- Verano: Julio, Agosto y Septiembre.
- Otoño: Octubre, Noviembre y Diciembre.

También se realizan agrupaciones por parejas de años para tratar de ver si existen coincidencias entre unos y otros y si es posible crear modelos para un año conociendo lo que ocurrió en el anterior. Por lo tanto, si se tiene la información de 2014 a 2017, se generan 24 muestras. En la siguiente tabla (*tabla 3.1*) puede verse a modo de ejemplo la división que se ha hecho de la muestra para la pareja de años 2014-2015. De igual modo se ha hecho para 2015-2016 y 2016-2017.

Muestras	Tipo de día	Estación	Parejas de años
1	Laboral	Invierno	2014-2015
2	Festivo		
3	Laboral	Primavera	
4	Festivo		
5	Laboral	Verano	
6	Festivo		
7	Laboral	Otoño	
8	Festivo		

Tabla 3.1 - 8 primeras divisiones de la muestra principal

Para cada una de las 24 muestras se han realizado 11 simulaciones. La primera ha sido siempre la clasificación según el algoritmo *ZeroR* para tener un primer valor con el que comparar el resto de resultados. De las 10 simulaciones restantes, 5 corresponden a pruebas realizadas con árboles de decisión (*J48*) y las otras 5 a redes neuronales (*MultilayerPerceptron*). Para cada uno de los dos métodos se han realizado 5 clasificaciones con distintas configuraciones. Para la primera se ha utilizado la opción *Use full training set*, con la que se puede ver cuál sería un porcentaje muy optimista de aciertos en la predicción. También se ha utilizado la validación cruzada con 10 divisiones y las tres restantes corresponden a la división de un porcentaje de la muestra para entrenamiento y el resto para testeo. De estos tres últimos casos, dos se han llevado a cabo sin ordenar la muestra y con divisiones de 50% y 66% para entrenamiento y el último caso ha sido el de la división del 50% de la muestra para entrenamiento manteniendo el orden de los datos. Esto último, corresponde a la realización de un modelo con los datos de un año y aplicarlo al siguiente.

En la siguiente tabla (*tabla 3.2*), se pueden ver los resultados de las simulaciones comentadas. Se ha definido un código de colores para poder extraer conclusiones de un modo más visual y el cual se incluye también en la imagen a modo de leyenda, siendo:

- Verde: resultados con porcentajes de aciertos mayores al 90%
- Azul: resultados con porcentajes de aciertos entre 85% y 90%
- Amarillo: resultados con porcentajes de aciertos entre 80% y 85%
- Naranja: resultados con porcentajes de aciertos entre 75% y 80%
- Rojo: resultados con porcentajes de aciertos menores al 75%

Como puede verse, las filas correspondientes a las simulaciones donde se ha utilizado la opción de *Use full training set*, son todas verdes, lo cual era previsible y debe tenerse en cuenta que son porcentajes muy optimistas. Tanto para las simulaciones donde no se ha mantenido el orden como para las de la validación cruzada, se encuentran resultados más intermedios, siendo todos los colores entre rojo y amarillo. Para la simulación del 50% con la muestra ordenada se obtienen los peores resultados de todos, quedando incluso lejos la mayor parte de veces del clasificador *ZeroR*.

Sentido del Desvío del Sistema									
2014-2015		14/15 INV Laboral	14/15 INV Festivo	14/15 PRI Laboral	14/15 PRI Festivo	14/15 VER Laboral	14/15 VER Festivo	14/15 OTO Laboral	14/15 OTO Festivo
50% Con orden	Red Neuronal	60,2865	47,0305	58,3333	55,7692	63,0682	51,6026	55,303	60,48
	Árbol de Decisión	57,2917	43,8202	54,1026	45,0321	62,4369	65,2244	57,5758	54,24
50% Sin orden	Red Neuronal	77,2786	78,6517	75,3205	78,0449	78,6616	76,6026	71,7172	81,28
	Árbol de Decisión	72,0703	73,0337	73,4615	75,3205	73,4217	77,8846	71,4015	73,92
66% Sin orden	Red Neuronal	76,1494	75,4717	74,8351	82,0755	80,5014	82,783	72,4234	80,2353
	Árbol de Decisión	74,7126	76,4151	72,0075	74,2925	74,6518	79,4811	69,3593	76,00
Cross Validation 10 folds	Red Neuronal	78,1576	80,7384	78,3013	81,5705	81,5972	81,891	75,00	83,68
	Árbol de Decisión	76,6602	76,9663	75,8013	75,8013	78,5669	77,484	73,3586	81,52
training set	Red Neuronal	94,4661	98,4751	93,2372	98,4776	95,1389	98,3173	90,4356	97,92
	Árbol de Decisión	96,0286	96,7095	96,891	96,9551	96,8119	96,4744	97,1907	97,28
ZeroR		61,4258	56,0995	67,2756	68,3494	67,4558	72,5962	63,9836	69,92
2015-2016		15/16 INV Laboral	15/16 INV Festivo	15/16 PRI Laboral	15/16 PRI Festivo	15/16 VER Laboral	15/16 VER Festivo	15/16 OTO Laboral	15/16 OTO Festivo
50% Con orden	Red Neuronal	58,7855	52,809	56,0256	57,3718	56,3131	56,891	51,3995	48,8226
	Árbol de Decisión	56,137	55,8587	52,7564	55,4487	53,2197	47,2756	50,5725	54,0031
50% Sin orden	Red Neuronal	73,2558	78,1701	76,3462	81,25	74,7475	75,9615	71,6921	78,179
	Árbol de Decisión	73,062	77,2071	71,7949	77,8846	72,4116	70,3526	70,3562	74,7253
66% Sin orden	Red Neuronal	77,4929	84,1981	77,7568	82,5472	77,1588	79,2453	68,8494	77,5982
	Árbol de Decisión	75,0237	75,2358	73,7983	79,2453	71,8663	74,0566	70,5332	74,1339
Cross Validation 10 folds	Red Neuronal	76,1305	81,4607	77,7885	82,2917	78,851	78,2853	73,4415	79,7488
	Árbol de Decisión	75,6783	76,565	75,3846	79,1667	75,947	77,7244	72,5191	77,4725
training set	Red Neuronal	91,4083	98,9567	93,7179	98,2372	92,6768	92,6768	90,4262	97,5667
	Árbol de Decisión	95,5749	96,7897	96,6346	96,0737	97,1907	96,6346	96,374	96,3893
ZeroR		62,7907	61,7175	66,3141	66,9872	59,4066	63,7821	61,3232	58,9482
2016-2017		16/17 INV Laboral	16/17 INV Festivo	16/17 PRI Laboral	16/17 PRI Festivo	16/17 VER Laboral	16/17 VER Festivo	16/17 OTO Laboral	16/17 OTO Festivo
50% Con orden	Red Neuronal	61,0897	47,6268	59,359	63,141	53,5623	63,141	56,9231	51,4638
	Árbol de Decisión	50,9615	58,1015	59,2308	55,9295	53,7532	50,4717	55,2564	51,926
50% Sin orden	Red Neuronal	74,6795	76,1047	71,3462	76,7628	73,5369	75,4717	72,0513	73,1895
	Árbol de Decisión	68,9744	69,7218	70,00	67,9487	68,5751	72,327	65,9615	69,9538
66% Sin orden	Red Neuronal	74,8351	77,8313	71,9133	77,3585	76,0524	79,1667	70,9708	80,4989
	Árbol de Decisión	72,8558	68,8795	70,311	68,1604	68,7558	74,3056	69,8398	72,5624
Cross Validation 10 folds	Red Neuronal	75,6731	80,0327	77,0833	80,0481	77,0038	79,6348	73,7821	80,0462
	Árbol de Decisión	74,6795	74,9591	75,1282	75,5609	75,5725	75,0786	72,7244	77,4268
training set	Red Neuronal	92,3718	98,6907	91,5064	97,8365	93,1298	98,4277	92,7885	97,6117
	Árbol de Decisión	96,5064	96,3993	95,7372	96,1538	96,374	95,5189	96,6026	97,1495
ZeroR		53,4295	53,1915	64,1987	60,1763	51,2723	51,4937	61,1538	52,5424

Mayor del 90 %
Entre 85 % y 90 %
Entre 80 % y 85 %
Entre 75 % y 80 %
Menor que 75 %

Figura 3.1 - Resultados del primer estudio para los desvíos del sistema

Aun no teniendo unos resultados muy positivos, sí que es posible extraer algunas ideas. La primera, es que parece complicado obtener un modelo de predicción para un año con los datos del año anterior. Es posible que probando otro tipo de agrupación de los datos sería posible mejorar en cierta medida los resultados. No obstante los resultados son tan negativos que no parece posible conseguir una mejora significativa.

Una vez desordenada la muestra y con el mismo porcentaje de división entrenamiento-testeo, ya se ve como mejoran en gran medida los porcentajes. Por ejemplo, en los días festivos de primavera de la pareja de años 2015-2016 y utilizando redes neuronales, se obtiene un 57.37% con la muestra ordenada y un 81.25% con la muestra sin ordenar. Esto puede indicar que exista una dependencia variable entre los desvíos y los atributos con los que se hace la predicción. Con esto se quiere decir que un atributo que resulta ser muy determinante en un año, puede no serlo al siguiente. Al aumentar el porcentaje de entrenamiento también suele mejorar la predicción y puede verse como los mejores resultados corresponden a la validación cruzada con diez divisiones, ya que se entrena con un 90% y se testea con el 10% restante, obteniendo al final una media aritmética de los aciertos de las diez divisiones.

También se puede comprobar que las redes neuronales suelen tener resultados más positivos que los árboles de decisión. Esto es algo que ya se vio en la metodología con la muestra de prueba y se corrobora ahora con un estudio más extenso. También se ha de decir que los tiempos de computación son muy superiores con las redes neuronales y las cantidades de datos que se manejan. En ninguno de los dos clasificadores se les ha modificado ninguno de los parámetros de configuración.

Por último, de manera general puede decirse que los resultados para las muestras correspondientes a días festivos son mejores que los de los días laborales. Esto puede deberse a simplemente a que tanto sábados como domingos, son días de caracteres similares y se puede estar haciendo un estudio más acotado. Es posible que se obtengan mejores resultados si en lugar de obtener una sola muestra para días laborales, se obtienen cinco (una para cada día de la semana de lunes a viernes).

Por otro lado, se piensa en las variables de las cuales son más dependientes los desvíos del sistema. En el estudio inicial de partida se vio que el participante del mercado que más desvíos cometía y el que solía definir el desvío global, era la demanda. También se vio que tanto eólica como fotovoltaica cometían desvíos pero de manera más discreta, no tanto para la eólica pero sí para la fotovoltaica. Por ello, se piensa que es posible predecir más fácilmente el desvío de la demanda que el desvío del sistema, al ser este último una composición de desvíos de distintos participantes. Si se hace una buena predicción del sentido al que se desviará la demanda, se tendrá con gran probabilidad el sentido del desvío global del sistema. Por lo tanto, se realiza el mismo estudio que se ha hecho para el sentido del desvío del sistema y la misma clasificación por colores según porcentaje de aciertos (*figura 3.2*).

Sentido del Desvío de la Demanda

2014-2015		14/15 INV Laboral	14/15 INV Festivo	14/15 PRI Laboral	14/15 PRI Festivo	14/15 VER Laboral	14/15 VER Festivo	14/15 OTO Laboral	14/15 OTO Festivo
50% Con orden	Red Neuronal	60,026	59,716	60,027	59,717	60,028	59,718	60,029	59,719
	Árbol de Decisión	48,763	57,785	53,4615	83,0449	60,4798	70,3526	49,9369	56,8
50% Sin orden	Red Neuronal	80,143	78,17	77,6282	76,6026	86,9949	80,4487	76,452	83,68
	Árbol de Decisión	75,651	71,5891	72,7564	70,9936	82,6389	75,9615	71,654	82,08
66% Sin orden	Red Neuronal	79,7893	80,6604	75,2121	80,8962	86,1653	83,2547	78,8301	84,7059
	Árbol de Decisión	77,682	73,82	74,1753	71,934	81,8013	79,4811	74,9304	82,1176
Cross Validation 10 folds	Red Neuronal	81,0221	83,5474	79,1026	79,9679	88,9205	84,6955	82,7724	79,1035
	Árbol de Decisión	80,013	78,0899	76,3462	74,5192	84,4066	82,7724	75,8838	85,68
training set	Red Neuronal	95,0195	99,0369	95,8974	99,2788	98,5164	98,7981	92,9609	97,84
	Árbol de Decisión	96,1263	96,87	96,2821	96,1538	97,4432	97,9968	95,8649	97,2
ZeroR		59,5703	54,655	55,8333	52,8846	66,0669	71,234	53,125	67,92
2015-2016		**15/16 INV Laboral**	**15/16 INV Festivo**	**15/16 PRI Laboral**	**15/16 PRI Festivo**	**15/16 VER Laboral**	**15/16 VER Festivo**	**15/16 OTO Laboral**	**15/16 OTO Festivo**
50% Con orden	Red Neuronal	50,1344	55,8587	50,7692	52,7244	61,4268	63,3013	56,3613	65,3061
	Árbol de Decisión	58,7209	49,4382	54,7436	48,3974	64,5202	63,4615	50,9542	64,9922
50% Sin orden	Red Neuronal	78,2946	83,1461	79,5513	81,4103	80,8712	87,3397	75,5725	79,9058
	Árbol de Decisión	77,1318	77,0465	74,1667	77,7244	77,8409	79,4872	74,3003	78,022
66% Sin orden	Red Neuronal	81,8613	83,7264	83,3176	81,3679	82,8227	86,0849	78,2975	82,679
	Árbol de Decisión	78,1576	81,1321	78,5108	77,3585	80,5014	80,8962	75,7717	78,5219
Cross Validation 10 folds	Red Neuronal	81,8152	84,1091	80,0161	81,6506	79,7756	88,8622	76,6221	83,752
	Árbol de Decisión	78,6499	80,0161	79,7756	79,0064	80,8712	83,8141	77,0674	80,1413
training set	Red Neuronal	96,6408	99,3579	95,641	98,6378	98,2008	99,4391	92,4936	99,0581
	Árbol de Decisión	96,8346	96,3082	97,5	95,9135	97,4432	97,9167	96,3104	98,1162
ZeroR		57,8811	62,9213	52,5321	56,5705	56,7866	65,3045	50,6361	63,8932
2016-2017		**16/17 INV Laboral**	**16/17 INV Festivo**	**16/17 PRI Laboral**	**16/17 PRI Festivo**	**16/17 VER Laboral**	**16/17 VER Festivo**	**16/17 OTO Laboral**	**16/17 OTO Festivo**
50% Con orden	Red Neuronal	67,1154	48,9362	63,7179	62,9808	51,2087	61,6352	60,8974	63,7904
	Árbol de Decisión	46,2179	46,4812	60,00	61,6987	49,8092	42,1384	57,8205	63,3282
50% Sin orden	Red Neuronal	80,3205	81,6694	83,2051	84,7756	80,7888	81,761	74,8718	83,2049
	Árbol de Decisión	78,7179	78,0687	79,4872	79,1667	72,7099	78,3019	74,4872	78,5824
66% Sin orden	Red Neuronal	83,6946	84,0964	81,5269	88,4434	82,0393	89,3519	76,9086	85,4875
	Árbol de Decisión	82,9406	77,8313	80,6786	83,4906	75,8653	82,4074	76,0603	79,1383
Cross Validation 10 folds	Red Neuronal	83,3974	83,3061	82,5962	85,5769	83,8422	87,9717	77,6282	86,2096
	Árbol de Decisión	81,1859	80,8511	81,1218	82,1314	79,0076	82,5472	77,4679	83,5901
training set	Red Neuronal	96,5064	98,9362	95,8013	98,7179	96,4695	99,0566	93,3654	99,4607
	Árbol de Decisión	97,4359	97,545	97,3077	96,3942	96,4695	98,1918	96,7949	97,7658
ZeroR		54,0385	55,0736	53,9744	53,3654	51,0178	51,9654	53,3974	50,2311

Mayor del 90 %
Entre 85 % y 90 %
Entre 80 % y 85 %
Entre 75 % y 80 %
Menor que 75 %

Figura 3.2 - Resultados del primer estudio para los desvíos de la demanda

Una vez se observan los resultados, se ve que es cierta la suposición de que los desvíos de la demanda son más fácilmente predecibles, al ser estos una componente de los desvíos del sistema y al depender directamente del comportamiento de la demanda. Se vuelven a extraer las mismas conclusiones que en el anterior estudio pero con porcentajes mayores. Como se

puede ver en la imagen, ya aumenta el número de casillas azules las cuales dan porcentajes cercanos al 90% de aciertos. Por otro lado, resulta también curioso que tanto en los desvíos de la demanda como en los del sistema se han obtenido las peores predicciones para los días laborales de los meses de otoño, mientras que los mejores resultados se tienen en los meses de verano y primavera y más aún si se miran días festivos. Aun así, sigue sin parecer posible el obtener un modelo con los datos de un año para aplicarlo al siguiente.

Al igual que se ha hecho con la demanda, se realiza el mismo estudio para los desvíos de la energía eólica. En la siguiente imagen (*figura 3.3*) puede verse el resultado de la simulación.

Sentido del Desvío Eólico

2014-2015		14/15 INV Laboral	14/15 INV Festivo	14/15 PRI Laboral	14/15 PRI Festivo	14/15 VER Laboral	14/15 VER Festivo	14/15 OTO Laboral	14/15 OTO Festivo
50% Con orden	Red Neuronal	53,71	57,78	52,50	52,88	51,45	51,44	60,98	49,92
	Árbol de Decisión	50,00	51,36	51,73	51,76	53,54	51,92	51,58	54,88
50% Sin orden	Red Neuronal	72,46	76,24	68,91	76,73	65,34	72,87	68,96	70,40
	Árbol de Decisión	73,44	70,47	65,32	72,87	64,58	69,34	73,38	71,52
66% Sin orden	Red Neuronal	72,06	80,19	70,90	78,07	68,25	79,48	72,42	75,94
	Árbol de Decisión	72,54	76,18	69,96	72,88	66,67	72,41	73,82	74,06
Cross Validation 10 folds	Red Neuronal	74,54	81,94	71,44	80,45	70,90	79,65	75,06	76,32
	Árbol de Decisión	76,89	78,57	71,15	76,45	69,13	74,51	77,43	75,36
training set	Red Neuronal	90,66	98,154	87,15	98,64	88,92	97,11	88,16	97,60
	Árbol de Decisión	96,615	97,27	96,15	96,00	95,17	96,71	97,03	96,64
ZeroR		44,01	53,13	38,40	32,69	43,37	49,04	65,97	55,04
2015-2016		**15/16 INV Laboral**	**15/16 INV Festivo**	**15/16 PRI Laboral**	**15/16 PRI Festivo**	**15/16 VER Laboral**	**15/16 VER Festivo**	**15/16 OTO Laboral**	**15/16 OTO Festivo**
50% Con orden	Red Neuronal	52,33	55,54	51,79	51,28	55,56	50,00	55,28	47,57
	Árbol de Decisión	55,88	59,23	55,06	54,97	52,59	48,24	55,47	44,43
50% Sin orden	Red Neuronal	69,46	73,79	73,65	73,88	68,12	75,12	72,71	73,63
	Árbol de Decisión	70,05	72,83	70,00	68,75	65,40	71,11	67,37	64,99
66% Sin orden	Red Neuronal	72,84	78,96	75,21	71,93	71,24	79,72	76,24	69,75
	Árbol de Decisión	74,17	77,78	74,27	75,00	65,68	73,11	70,53	66,05
Cross Validation 10 folds	Red Neuronal	73,29	80,49	73,94	76,52	71,31	81,33	76,53	77,16
	Árbol de Decisión	74,87	77,68	75,77	74,28	71,31	74,83	75,86	76,05
training set	Red Neuronal	88,44	96,87	90,24	97,92	85,77	98,78	90,2672	96,938
	Árbol de Decisión	96,32	95,59	96,19	95,59	96,21	96,875	96,0242	95,84
ZeroR		60,85	48,48	70,71	57,37	50,82	52,40	64,38	47,57
2016-2017		**16/17 INV Laboral**	**16/17 INV Festivo**	**16/17 PRI Laboral**	**16/17 PRI Festivo**	**16/17 VER Laboral**	**16/17 VER Festivo**	**16/17 OTO Laboral**	**16/17 OTO Festivo**
50% Con orden	Red Neuronal	56,03	58,10	48,78	51,44	54,01	47,01	51,73	52,54
	Árbol de Decisión	53,3333	54,8282	55,9615	52,5641	53,4987	50,00	53,7179	45,4545
50% Sin orden	Red Neuronal	74,50	74,96	75,06	73,52	71,12	79,56	72,58	75,96
	Árbol de Decisión	68,6538	69,8854	70,5128	69,7115	66,285	69,8113	70,1923	70,5701
66% Sin orden	Red Neuronal	72,32	81,97	76,15	74,47	72,97	79,91	72,79	79,37
	Árbol de Decisión	70,1225	78,5542	75,0236	65,3302	67,7268	71,5278	74,4581	69,3878
Cross Validation 10 folds	Red Neuronal	76,70	81,01	76,19	77,40	73,44	79,72	75,32	79,98
	Árbol de Decisión	75,0321	77,9051	75,2564	72,6763	69,5611	73,978	76,0577	76,4253
training set	Red Neuronal	91,4423	98,6088	91,0897	98,5577	91,9847	98,6635	91,5064	97,8428
	Árbol de Decisión	96,3462	97,7087	96,3141	95,4327	94,5611	97,7201	96,5064	95,9168
ZeroR		57,9808	57,365	69,359	60,9776	51,7176	52,044	62,3397	53,698

Mayor del 90 %
Entre 85 % y 90 %
Entre 80 % y 85 %
Entre 75 % y 80 %
Menor que 75 %

Figura 3.3 - Resultados del primer estudio para los desvíos eólicos

Vuelve a darse el mismo problema que para los anteriores estudios, donde no resultaba posible realizar un modelo con los datos de un año y aplicarlo al siguiente. También se repite el mismo comportamiento donde a medida que se aumenta el tamaño de la muestra de entrenamiento se obtienen mejores resultados. Aun así, los porcentajes de aciertos en las predicciones para los desvíos de la energía eólica son muy inferiores a los que se han obtenido en los desvíos de la demanda. Esto puede tener varias explicaciones. Una de ellas es que los desvíos eólicos pueden no ser fuertemente dependientes de los atributos seleccionados para realizar la predicción, aunque eso a priori no es posible saberlo. Otra explicación puede ser que durante los últimos años los desvíos eólicos, a diferencia de los desvíos de la demanda que suelen tener un claro comportamiento repetitivo año tras año (excepto en 2015), han tenido un comportamiento irregular, con lo que realizar una predicción ha podido resultar más difícil. Por ejemplo, algunos de estos comportamientos irregulares se conoce que fueron creados por cambios retributivos hacia las renovables y estos son aspectos que no se han tenido en cuenta a la hora de realizar la

predicción. Toda esta información no es posible verla en las tablas de resultados expuestas y son conclusiones extraídas del proyecto de referencia donde se realizó el análisis de los desvíos del sistema. A continuación se expone una imagen (*figura 3.4*) extraída de dicho proyecto donde se ve la tendencia diaria de los desvíos a subir de la generación eólica durante los años 2014, 2015 y 2016.

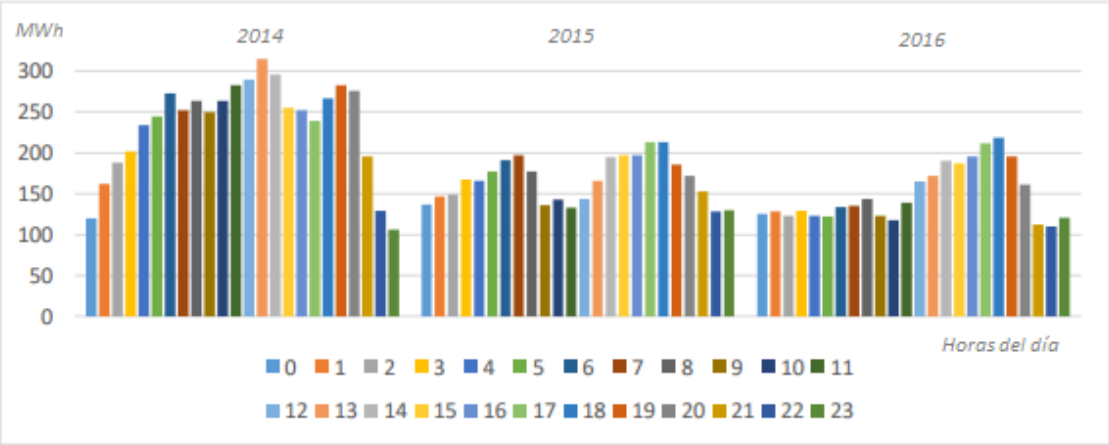


Figura 3.4 - Energía Media Horaria de los Desvíos a Subir de la Generación Eólica en MWh 2014,2015 y 2016

Aun así, tal como se ha comentado anteriormente, la demanda también tuvo un comportamiento distinto durante el año 2015 (*figura 3.5*) y no por ello se ven penalizadas las predicciones de este año con respecto a las de los otros. Con lo que puede suponerse que independientemente del comportamiento regular o no regular de estas variables, finalmente lo importante es contar con los atributos de las cuales están son dependientes. Así, si alguno de estos atributos cambia por razones aleatorias, igualmente se podrá predecir el comportamiento aparentemente aleatorio del desvío.

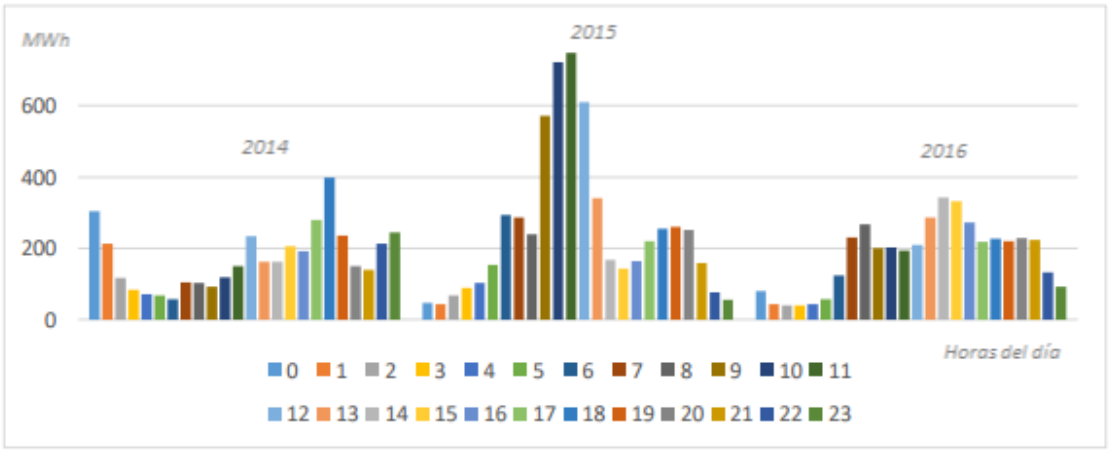


Figura 3.5 - Energía Media Horaria de los Desvíos a Subir de la Demanda en MWh 2014,2015 y 2016

Por último, se realiza el estudio para los desvíos de la generación fotovoltaica. En la siguiente imagen (*figura 3.6*) pueden verse los resultados.

Sentido del Desvío Fotovoltaico									
2014-2015		14/15 INV Laboral	14/15 INV Festivo	14/15 PRI Laboral	14/15 PRI Festivo	14/15 VER Laboral	14/15 VER Festivo	14/15 OTO Laboral	14/15 OTO Festivo
50% Con orden	Red Neuronal	67,58	53,77	58,08	56,73	59,85	62,50	61,87	60,00
	Árbol de Decisión	71,09	74,16	65,00	67,79	64,52	68,75	65,34	49,92
50% Sin orden	Red Neuronal	79,10	79,45	77,76	78,85	76,40	77,88	78,99	76,80
	Árbol de Decisión	83,59	80,74	79,81	82,69	82,27	81,25	82,65	82,88
66% Sin orden	Red Neuronal	78,16	84,16	76,81	82,55	79,67	78,30	78,92	79,29
	Árbol de Decisión	82,38	84,63	80,58	83,96	83,10	86,79	83,84	83,76
Cross Validation 10 folds	Red Neuronal	83,43	82,02	78,97	82,37	79,51	80,93	80,75	79,12
	Árbol de Decisión	85,74	87,15	82,85	83,66	83,59	85,42	83,87	86,40
training set	Red Neuronal	96,35	99,04	94,00	98,47	94,48	98,15	93,87	98,16
	Árbol de Decisión	97,88	97,75	97,40	96,55	97,41	97,67	97,63	98,40
ZeroR		47,66	51,36	42,88	45,03	33,71	36,70	55,30	44,64
2015-2016		15/16 INV Laboral	15/16 INV Festivo	15/16 PRI Laboral	15/16 PRI Festivo	15/16 VER Laboral	15/16 VER Festivo	15/16 OTO Laboral	15/16 OTO Festivo
50% Con orden	Red Neuronal	55,56	52,65	55,06	50,74	58,14	44,87	65,97	57,46
	Árbol de Decisión	51,81	49,60	60,96	59,78	46,28	57,85	57,12	54,00
50% Sin orden	Red Neuronal	77,45	74,96	78,59	77,37	76,89	78,97	77,42	81,32
	Árbol de Decisión	82,04	79,29	78,65	77,85	81,00	82,18	79,39	83,99
66% Sin orden	Red Neuronal	79,56	80,61	78,13	76,65	76,72	76,89	78,65	79,68
	Árbol de Decisión	82,60	82,03	79,08	79,25	79,13	82,78	79,12	83,83
Cross Validation 10 folds	Red Neuronal	78,26	79,69	79,42	79,09	79,17	83,01	79,39	81,63
	Árbol de Decisión	83,46	84,11	82,53	80,77	82,95	84,13	82,95	84,38
training set	Red Neuronal	94,73	98,47	92,98	97,44	94,22	99,03	92,08	98,74
	Árbol de Decisión	97,12	97,51	96,15	97,12	97,47	97,19	96,78	97,02
ZeroR		54,97	49,76	64,23	63,78	58,08	67,95	58,02	39,87
2016-2017		16/17 INV Laboral	16/17 INV Festivo	16/17 PRI Laboral	16/17 PRI Festivo	16/17 VER Laboral	16/17 VER Festivo	16/17 OTO Laboral	16/17 OTO Festivo
50% Con orden	Red Neuronal	54,17	56,30	58,33	64,58	41,73	47,33	60,51	70,42
	Árbol de Decisión	41,92	49,43	56,35	64,74	52,93	55,66	67,12	68,10
50% Sin orden	Red Neuronal	77,95	72,67	74,55	76,24	74,24	79,40	75,71	80,28
	Árbol de Decisión	80,00	77,74	78,78	75,44	80,98	81,60	75,83	78,43
66% Sin orden	Red Neuronal	79,83	77,64	77,29	75,24	75,77	79,45	75,59	80,27
	Árbol de Decisión	81,43	76,44	79,92	76,89	82,79	81,52	76,44	78,68
Cross Validation 10 folds	Red Neuronal	79,23	79,70	76,31	75,89	78,31	81,29	77,82	82,36
	Árbol de Decisión	84,07	80,69	80,96	80,61	84,35	83,57	81,44	81,04
training set	Red Neuronal	93,365	98,527	90,77	97,276	91,54	97,95	91,731	98,00
	Árbol de Decisión	96,38	96,90	96,86	96,71	97,42	96,62	96,22	97,23
ZeroR		59,29	57,94	46,60	51,76	45,48	49,53	69,68	67,18

Mayor del 90 %
Entre 85 % y 90 %
Entre 80 % y 85 %
Entre 75 % y 80 %
Menor que 75 %

Figura 3.6 - Resultados del primer estudio para los desvíos fotovoltaicos

Aquí, al igual que con la demanda, vuelven a verse porcentajes de aciertos positivos, aunque del mismo modo, no parece posible el generar un modelo de un año para otro. Esto último va en contraposición con algunas conclusiones que se obtuvieron en el anterior proyecto. Precisamente, los desvíos fotovoltaicos se sucedían de manera tan similar año tras año (*figura 3.7*), que de la simple observación de las curvas de un año era posible desarrollar una estrategia de oferta para el año siguiente que reducía las pérdidas por desvíos en cerca de un 50% en la mayoría de casos. Probablemente, este caso de predicción sea tan sencillo que el incluir tal cantidad de atributos genere más problemas que soluciones, siendo además el desvío que menos interés tiene en ser predicho ya que, en porcentaje dentro de los desvíos del sistema, es el que menor presencia tiene si se le compara con los desvíos eólicos o de la demanda.

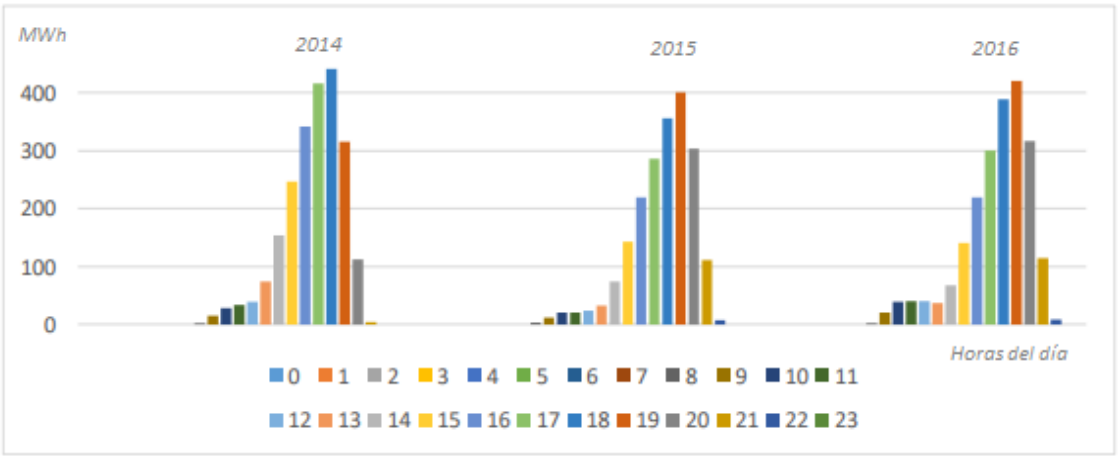


Figura 3.7 - Energía Media Horaria de los Desvíos a Subir de la Generación Fotovoltaica en MWh 2014,2015 y 2016

Una vez vistos los resultados de las clasificaciones anteriores, se decide comprobar si el uso de la herramienta de selección de atributos mejora o no los porcentajes de aciertos. La selección de atributos se hará para el estudio de los desvíos de la demanda, dado que los resultados de esta predicción han sido los más positivos. Para acotar el estudio, se decide realizar la selección solo para las 8 muestras de la pareja de años de 2016 y 2017. Para realizar la selección se utilizará el método de evaluación *ClassifierSubsetEval* con *GreedyStepwise* como método de búsqueda. En este caso, al ser un método *wrapper*¸ habrá que incluir un clasificador base. Se selecciona una red neuronal y se decide utilizar la validación cruzada con 10 divisiones. En la siguiente imagen (*figura 3.8*) se puede ver el resultado para las 8 muestras y los atributos ordenados de mayor a menor presencia en las selecciones realizadas.

Porcentajes de Peso en Sentido del Desvío de la Demanda. ClassifierSubsetEval. GreedyStepwise. Red neuronal

Variables	16/17 INV Laboral	16/17 INV Festivo	16/17 PRI Laboral	16/17 PRI Festivo	16/17 VER Laboral	16/17 VER Festivo	16/17 OTO Laboral	16/17 OTO Festivo	Media
mes	100	50	70	70	60	90	60	30	66.25
TemperaturaMadrid	70	40	100	60	100	50	90	10	65
hora	60	30	100	90	50	20	70	100	65
NuclearPVP	30	80	50	60	50	100	60	50	60
TemperaturaGibraltar	90	80	40	10	70	40	50	100	60
HumedadGibraltar	60	70	60	40	100	30	40	50	56.25
PresionMadrid	60	60	40	90	40	40	10	70	51.25
día	80	70	60	20	80	30	30	30	50
PresionGibraltar	60	50	70	80	50	10	30	40	48.75
FotovoltaicaPVP	30	60	30	20	90	80	70	0	47.5
HumedadMadrid	60	30	70	50	50	20	40	60	47.5
VientoGibraltar	30	30	30	40	50	80	40	50	43.75
HidraulicaPVP	70	30	80	70	30	30	30	0	42.5
InterconexionesPVP	30	60	20	60	50	20	70	30	42.5
EnlaceBalearesPVP	50	100	60	30	30	20	30	10	41.25
CarbonPVP	40	30	50	30	80	30	20	40	40
RegSecundariaSubir	20	30	70	40	40	30	70	20	40
TemperaturaVigo	30	80	40	20	30	40	30	40	38.75
CogeneracionPVP	30	20	30	70	60	30	20	30	36.25
CicloCombinadoPVP	40	40	30	40	50	30	40	10	35
HumedadVigo	50	20	40	20	20	40	20	60	33.75
DemandaPVP	10	20	40	40	50	60	30	0	31.25
TermosolarPVP	40	10	20	30	70	30	40	0	30
PrecioMercadoDiario	30	0	40	30	70	10	30	20	28.75
EolicaPVP	60	30	20	10	20	0	50	30	27.5
RegSecundariaBajar	20	20	50	30	30	10	10	10	22.5
BombeoPVP	20	40	10	30	20	30	0	20	21.25
VientoMadrid	30	50	20	30	10	20	10	0	21.25
VientoVigo	10	20	20	10	20	40	0	50	21.25
TurbinacionBombeoPVP	50	0	40	0	10	20	20	0	17.5
ReservaPotencia	20	0	10	10	0	0	20	0	7.5
PresionVigo	0	0	0	0	0	0	0	0	0

Figura 3.8 – Selección mediante validación cruzada y método wrapper

Al ser una validación cruzada con diez divisiones, cada atributo presentará un porcentaje en cada columna. Este porcentaje corresponde al número de veces que fue seleccionado dentro del grupo de atributos elegidos en cada una de las divisiones. Es decir, si un atributo aparece con un 60% significará que en 6 de las 10 divisiones formo parte del grupo seleccionado. Es posible ver que los atributos varían su eficacia según la estación o el día en el que se encuentren. Al realizar la media de todo el año, hay un grupo de 5 atributos que presentan valores iguales o superiores al 60%. En cambio en el caso concreto del atributo *PresionVigo*, no ha sido seleccionado en ninguna de las columnas ni en ninguna de las divisiones realizadas con lo que es un claro atributo a ser descartado. En la siguiente imagen (*figura 3.9*), puede verse el efecto que crea el realizar la clasificación reduciendo el número de atributos escalonadamente según su porcentaje de aparición. Se ha utilizado únicamente la muestra para días laborales de invierno.

Clasificación de los Desvíos de la Demanda. ClassifierSubsetEval. GreedyStepwise. Red neuronal

2016-2017		16/17 INV Laboral	16/17 INV Laboral	16/17 INV Laboral	16/17 INV Laboral	16/17 INV Laboral	16/17 INV Laboral	16/17 INV Laboral	16/17 INV Laboral
50% Con orden	Red Neuronal	67,1154	58,3333	59,4231	63,8462	58,4615	66,6026	60,8974	66,3462
	Árbol de Decisión	46,2179	46,2179	46,2821	50,5128	51,4103	51,4744	51,2179	60,3205
50% Sin orden	Red Neuronal	80,3205	81,9231	82,8846	79,9359	76,7949	80,3205	79,6154	76,0897
	Árbol de Decisión	78,7179	78,7179	78,3333	78,3974	77,3718	78,5256	75,00	76,7308
66% Sin orden	Red Neuronal	83,6946	81,0556	83,3176	81,7154	81,4326	80,3016	80,3959	76,7201
	Árbol de Decisión	82,9406	82,9406	83,0349	81,1499	80,3016	82,8464	80,3016	75,9661
Cross Validation 10 folds	Red Neuronal	83,3974	84,3269	83,5897	83,3974	82,9167	81,5064	78,3013	75,5769
	Árbol de Decisión	81,1859	81,1859	81,1218	81,5705	80,9936	82,5641	82,6603	79,6795
training set	Red Neuronal	96,5064	96,2179	95,8974	92,9167	88,4936	85,7372	84,2628	76,2179
	Árbol de Decisión	97,4359	97,4359	97,2756	96,3782	96,6346	96,5385	95,641	88,4936
ZeroR		54,0385	54,0385	54,0385	54,0385	54,2884	54,0385	54,0385	54,0385
Selección de atributos		Muestra completa	Eliminando menores de 10	Eliminando menores de 20	Eliminando menores de 30	Eliminando menores de 40	Eliminando menores de 50	Eliminando menores de 60	Eliminando menores de 70

Mayor del 90 %
Entre 85 % y 90 %
Entre 80 % y 85 %
Entre 75 % y 80 %
Menor que 75 %

Figura 3.9 - Clasificación mediante selección con validación cruzada, wrapper y red neuronal

Es posible ver como los porcentajes van variando en cierta medida según se van eliminando los atributos menos importantes, aunque no es posible ver un sentido definido para estas fluctuaciones. En algunos casos parece que eliminar atributos mejora la predicción y en otras que la empeora. En cualquier caso, no se detecta una mejoría o empeoramiento significativo hasta que se empiezan a eliminar los atributos con porcentajes menores a 60-70%, donde ya sí que se ve un claro empeoramiento de los resultados. No obstante, parece que los mejores resultados aparecen con la muestra completa o simplemente eliminado los pocos atributos que nunca fueron seleccionados. Es aquí donde se debe poner en balance si conviene reducir en cierta medida el porcentaje de aciertos pero conseguir menores tiempos de computación. En cualquier caso, parece que el añadir atributos lo mínimo que hace es mejorar y no empeorar el resultado de la clasificación.

Teniendo en cuenta el coste computacional que requiere realizar la clasificación anterior, donde se ha utilizado validación cruzada con diez divisiones y redes neuronales, no parece que merezca la pena dedicar tanto esfuerzo para obtener los resultados que se han obtenido. Se probará a realizar la misma clasificación pero utilizando la muestra completa sin divisiones, con lo cual se espera reducir los tiempos de computación. En la siguiente imagen (*figura 3.10*), se pueden ver los resultados para las 8 divisiones que se han hecho de la pareja de años 2016-2017, donde se han marcado con un *0* los atributos que no han sido seleccionados y con un *1* los que sí lo han sido. Finalmente se obtiene la suma de las veces que cada atributo ha sido seleccionado en todo el año y se ordenan de mayor a menor aparición.

Selección de grupos de atributos. ClassifierSubsetEval. GreedyStepwise. MultilayerPerceptron. UseFullTrainingSet

Variables	16/17 INV Laboral	16/17 INV Festivo	16/17 PRI Laboral	16/17 PRI Festivo	16/17 VER Laboral	16/17 VER Festivo	16/17 OTO Laboral	16/17 OTO Festivo	Media
TemperaturaGibraltar	1	1	1	1	1	0	1	1	7
VientoGibraltar	0	1	1	1	1	1	1	1	7
TemperaturaMadrid	1	0	1	1	1	1	1	0	6
día	1	1	1	0	1	1	1	0	6
hora	1	0	1	1	0	1	1	1	6
CarbonPVP	1	0	0	1	1	1	1	0	5
InterconexionesPVP	1	1	0	1	1	0	1	0	5
NuclearPVP	0	1	1	0	1	1	1	0	5
EnlaceBalearesPVP	1	1	1	0	0	1	0	0	4
CicloCombinadoPVP	1	0	1	1	0	1	0	0	4
FotovoltaicaPVP	0	1	0	1	1	1	0	0	4
HidraulicaPVP	1	0	1	1	0	1	0	0	4
RegSecundariaSubir	0	0	1	1	0	0	1	1	4
HumedadMadrid	0	0	0	1	1	0	1	1	4
PresionMadrid	0	0	1	1	0	1	1	0	4
HumedadGibraltar	0	1	0	0	1	1	1	0	4
mes	1	0	0	1	1	1	0	0	4
PrecioMercadoDiario	1	0	0	0	1	0	1	1	4
CogeneracionPVP	1	0	0	1	0	1	0	0	3
DemandaPVP	0	1	0	0	0	1	1	0	3
ReservaPotencia	1	0	0	0	0	1	1	0	3
TermosolarPVP	1	0	0	1	0	0	1	0	3
TurbinacionBombeoPVP	0	0	0	0	0	1	1	1	3
PresionGibraltar	1	0	0	1	0	0	0	1	3
BombeoPVP	0	0	0	0	0	1	1	0	2
RegSecundariaBajar	0	0	0	0	0	0	1	1	2
TemperaturaVigo	0	1	0	1	0	0	0	0	2
HumedadVigo	1	0	0	1	0	0	0	0	2
EolicaPVP	1	0	0	0	0	0	0	0	1
VientoMadrid	1	0	0	0	0	0	0	0	1
VientoVigo	0	0	0	0	0	0	1	0	1
PresionVigo	0	0	0	0	0	0	0	0	0

Figura 3.10 - Selección simple mediante método wrapper y redes neuronales

De nuevo vuelve a quedar el atributo *PresionVigo* en el último lugar y atributos como *TemperaturaGibraltar* o *TemperaturaMadrid*, siguen estando bien posicionados arriba. A continuación en la siguiente imagen (*figura 3.11*), se ve como queda la clasificación utilizando la anterior selección de atributos para las 8 muestras de la pareja de años seleccionada.

Clasificación de los Desvíos de la Demanda. ClassifierSubsetEval. GreedyStepwise. Redes Neuronales

2016-2017		16/17 INV Laboral	16/17 INV Laboral	16/17 INV Festivo	16/17 INV Festivo	16/17 PRI Laboral	16/17 PRI Laboral	16/17 PRI Festivo	16/17 PRI Festivo
50% Con orden	Red Neuronal	67,1154	56,7308	83,3974	38,1342	63,7179	58,7179	62,9808	55,609
	Árbol de Decisión	46,2179	56,0256	46,4812	57,4468	60,00	54,2949	61,6987	52,2436
50% Sin orden	Red Neuronal	80,3205	80,3846	81,6694	78,7234	83,2051	80,00	84,7756	85,4167
	Árbol de Decisión	78,7179	74,0385	78,0687	78,5597	79,4872	78,0128	79,1667	79,6474
66% Sin orden	Red Neuronal	83,6946	79,7361	84,0964	82,8916	81,5269	78,0396	88,4434	85,1415
	Árbol de Decisión	82,9406	79,8303	77,8313	76,1446	80,6786	77,0028	83,4906	78,5377
Cross Validation 10 folds	Red Neuronal	83,3974	82,4679	83,3061	82,9787	82,5962	79,0064	85,5769	86,0577
	Árbol de Decisión	81,1859	80,2564	80,8511	81,6694	81,1218	81,859	82,1314	84,1346
training set	Red Neuronal	96,5064	91,2179	98,9362	92,0622	95,8013	86,4744	98,7179	97,6763
	Árbol de Decisión	97,4359	96,5705	97,545	95,0082	97,3077	95,2885	96,3942	96,9551
ZeroR		54,0385	54,0385	55,0736	55,0736	53,9744	53,9744	53,3654	53,3654
2016-2017		**16/17 VER Laboral**	**16/17 VER Laboral**	**16/17 VER Festivo**	**16/17 VER Festivo**	**16/17 OTO Laboral**	**16/17 OTO Laboral**	**16/17 OTO Festivo**	**16/17 OTO Festivo**
50% Con orden	Red Neuronal	51,2087	57,0611	61,6352	61,6352	60,8974	59,5513	63,7904	64,0986
	Árbol de Decisión	49,8092	56,1069	42,1384	60,5346	57,8205	53,3974	63,3282	60,2465
50% Sin orden	Red Neuronal	80,7888	78,4987	81,761	82,2327	74,8718	70,1923	83,2049	76,5794
	Árbol de Decisión	72,7099	77,4809	78,3019	80,9748	74,4872	72,9487	78,5824	75,6549
66% Sin orden	Red Neuronal	82,0393	75,8653	89,3519	85,4167	76,9086	72,3845	85,4875	80,4989
	Árbol de Decisión	75,8653	77,362	82,4074	81,0185	76,0603	72,573	79,1383	78,2313
Cross Validation 10 folds	Red Neuronal	83,8422	79,8028	87,9717	85,9277	77,6282	74,1346	86,2096	81,8952
	Árbol de Decisión	79,0076	78,5305	82,5472	84,5912	77,4679	75,7692	83,5901	82,5116
training set	Red Neuronal	96,4695	86,7366	99,0566	97,7201	93,3654	86,7949	99,4607	91,2173
	Árbol de Decisión	96,4695	95,4835	98,1918	96,6981	96,7949	96,0897	97,7658	96,7643
ZeroR		51,0178	51,0178	51,9654	51,9654	53,3974	53,3974	50,2311	50,2311
Selección de atributos		Todos los atributos	Selección con Red Neuronal	Todos los atributos	Selección con Red Neuronal	Todos los atributos	Selección con Red Neuronal	Todos los atributos	Selección con Red Neuronal

Mayor del 90 %
Entre 85 % y 90 %
Entre 80 % y 85 %
Entre 75 % y 80 %
Menor que 75 %

Figura 3.11 - Clasificación mediante selección simple, wrapper y redes neuronales

Se observa que por lo general, la eliminación de atributos empeora los resultados pero no lo hace de manera excesiva. Aun así, si antes se eliminaban atributos de manera gradual porque

se disponía de unos porcentajes, ahora se ha hecho cogiendo solo los atributos seleccionados en cada una de las muestras, es decir, una selección de atributos todo-nada. También el tiempo de computación para obtener la selección ha sido significativamente más corto. A continuación se exponen los resultados de selección y clasificación utilizando el mismo último método pero con árboles de decisión como clasificador base (*figuras 3.12 y 3.13*).

Selección de grupos de atributos. ClassifierSubsetEval. GreedyStepwise. J48. UseFullTrainingSet

Variables	16/17 INV Laboral	16/17 INV Festivo	16/17 PRI Laboral	16/17 PRI Festivo	16/17 VER Laboral	16/17 VER Festivo	16/17 OTO Laboral	16/17 OTO Festivo	Media
NuclearPVP	1	0	1	1	1	1	1	1	7
EolicaPVP	1	0	1	1	1	0	1	1	6
CarbonPVP	1	1	0	0	1	0	1	1	5
PresionGibraltar	0	1	1	1	1	0	0	1	5
hora	0	1	1	1	1	0	1	0	5
BombeoPVP	1	0	0	1	1	0	1	0	4
CicloCombinadoPVP	0	1	1	0	0	0	1	1	4
DemandaPVP	1	0	1	0	1	0	1	0	4
FotovoltaicaPVP	1	0	0	0	0	1	1	1	4
VientoMadrid	1	1	0	1	0	0	1	0	4
PresionMadrid	1	1	1	0	1	0	0	0	4
HumedadGibraltar	1	1	0	1	0	0	1	0	4
día	1	0	1	0	1	0	1	0	4
CogeneracionPVP	0	0	0	0	1	1	0	1	3
HidraulicaPVP	1	0	0	1	0	0	1	0	3
InterconexionesPVP	0	1	0	1	1	0	0	0	3
TermosolarPVP	1	0	1	0	0	1	0	0	3
TurbinacionBombeoPVP	1	0	1	0	0	0	0	1	3
VientoGibraltar	0	1	1	0	1	0	0	0	3
mes	0	0	1	0	1	0	0	1	3
RegSecundariaSubir	0	0	0	0	0	1	0	1	2
TemperaturaGibraltar	0	0	0	0	0	1	0	1	2
TemperaturaVigo	0	0	0	0	1	0	1	0	2
HumedadVigo	1	0	0	0	0	1	0	0	2
PrecioMercadoDiario	0	0	0	0	1	0	1	0	2
RegSecundariaBajar	0	0	0	1	0	0	0	0	1
VientoVigo	0	0	0	0	0	0	1	0	1
EnlaceBalearesPVP	0	0	0	0	0	0	0	0	0
ReservaPotencia	0	0	0	0	0	0	0	0	0
TemperaturaMadrid	0	0	0	0	0	0	0	0	0
HumedadMadrid	0	0	0	0	0	0	0	0	0
PresionVigo	0	0	0	0	0	0	0	0	0

Figura 3.12 - Selección simple mediante método wrapper y árboles de decisión

Clasificación de los Desvíos de la Demanda. ClassifierSubsetEval. GreedyStepwise. Árbol de decisión

2016-2017		16/17 INV Laboral	16/17 INV Laboral	16/17 INV Festivo	16/17 INV Festivo	16/17 PRI Laboral	16/17 PRI Laboral	16/17 PRI Festivo	16/17 PRI Festivo
50% Con orden	Red Neuronal	67,1154	57,8205	83,3974	39,4435	63,7179	54,4872	62,9808	52,7244
	Árbol de Decisión	46,2179	51,7949	46,4812	45,1718	60,00	51,2821	61,6987	62,5
50% Sin orden	Red Neuronal	80,3205	74,1667	81,6694	69,8854	83,2051	71,6667	84,7756	77,4038
	Árbol de Decisión	78,7179	75,5769	78,0687	75,2864	79,4872	79,5513	79,1667	78,3654
66% Sin orden	Red Neuronal	83,6946	76,8143	84,0964	71,8072	81,5269	78,0396	88,4434	80,8962
	Árbol de Decisión	82,9406	77,2856	77,8313	76,1446	80,6786	78,8878	83,4906	77,3585
Cross Validation 10 folds	Red Neuronal	83,3974	77,2115	83,3061	72,8314	82,5962	75,4808	85,5769	79,3269
	Árbol de Decisión	81,1859	79,4551	80,8511	74,8773	81,1218	81,3782	82,1314	79,2468
training set	Red Neuronal	96,5064	81,7949	98,9362	82,3241	95,8013	79,7436	98,7179	89,0224
	Árbol de Decisión	97,4359	97,2436	97,545	96,9722	97,3077	97,0833	96,3942	97,9968
ZeroR		54,0385	54,0385	55,0736	55,0736	53,9744	53,9744	53,3654	53,3654
2016-2017		**16/17 VER Laboral**	**16/17 VER Laboral**	**16/17 VER Festivo**	**16/17 VER Festivo**	**16/17 OTO Laboral**	**16/17 OTO Laboral**	**16/17 OTO Festivo**	**16/17 OTO Festivo**
50% Con orden	Red Neuronal	51,2087	40,7761	61,6352	64,7799	60,8974	55,7692	63,7904	71,3405
	Árbol de Decisión	49,8092	48,2824	42,1384	68,239	57,8205	59,9359	63,3282	62,8659
50% Sin orden	Red Neuronal	80,7888	75,827	81,761	78,1447	74,8718	71,4744	83,2049	76,5794
	Árbol de Decisión	72,7099	74,4275	78,3019	76,1006	74,4872	71,0256	78,5824	78,2743
66% Sin orden	Red Neuronal	82,0393	76,8943	89,3519	74,0741	76,9086	69,8398	85,4875	82,0862
	Árbol de Decisión	75,8653	78,2039	82,4074	82,1759	76,0603	74,3638	79,1383	81,4059
Cross Validation 10 folds	Red Neuronal	83,8422	78,6896	87,9717	75,8648	77,6282	72,7564	86,2096	81,8952
	Árbol de Decisión	79,0076	79,771	82,5472	81,3679	77,4679	76,3141	83,5901	82,6656
training set	Red Neuronal	96,4695	86,1005	99,0566	77,8302	93,3654	79,9038	99,4607	90,755
	Árbol de Decisión	96,4695	97,7099	98,1918	96,305	96,7949	97,2436	97,7658	97,9199
ZeroR		51,0178	51,0178	51,9654	51,9654	53,3974	53,3974	50,2311	50,2311
Selección de atributos		Todos los atributos	Selección con Árbol de Decisión	Todos los atributos	Selección con Árbol de Decisión	Todos los atributos	Selección con Árbol de Decisión	Todos los atributos	Selección con Árbol de Decisión

Mayor del 90 %
Entre 85 % y 90 %
Entre 80 % y 85 %
Entre 75 % y 80 %
Menor que 75 %

Figura 3.13 - Clasificación mediante selección simple, wrapper y árbol de decisión

En este caso, sí que han empeorado de manera significativa los resultados de la clasificación. Por lo que se ve, dada la cantidad de datos de las muestras, la complejidad del estudio y con una reducción de los atributos, la red neuronal es capaz de obtener mejores resultados que los árboles de decisión. Lo que puede llegar a resultar curioso, es la relación de atributos que cada clasificador considera importantes. En la siguiente tabla (*tabla 3.2*), se comparan los rankings obtenidos en cada una de las tres clasificaciones con un ranking general obtenido de aplicar una media ponderada a los resultados obtenidos por cada atributo en cada una de las selecciones.

Porcentajes de Peso en Sentido del Desvío de la Demanda. ClassifierSubsetEval. GreedyStepwise. Red neuronal		**Selección de grupos de atributos. ClassifierSubsetEval. GreedyStepwise. MultilayerPerceptron. UseFullTrainingSet**		**Selección de grupos de atributos. ClassifierSubsetEval. GreedyStepwise. J48. UseFullTrainingSet**		**Media de cada atributo resultante de la ponderación de resultados de cada selección**	
mes	66.25	TemperaturaGibraltar	7	NuclearPVP	6	NuclearPVP	12
TemperaturaMadrid	65	VientoGibraltar	7	EolicaPVP	6	hora	11
hora	65	TemperaturaMadrid	6	CarbonPVP	5	CarbonPVP	10
NuclearPVP	60	día	6	PresionGibraltar	5	VientoGibraltar	10
TemperaturaGibraltar	60	hora	6	hora	5	día	10
HumedadGibraltar	56.25	CarbonPVP	5	BombeoPVP	4	TemperaturaGibraltar	9
PresionMadrid	51.25	InterconexionesPVP	5	CicloCombinadoPVP	4	CicloCombinadoPVP	8
día	50	NuclearPVP	5	DemandaPVP	4	FotovoltaicaPVP	8
PresionGibraltar	48.75	EnlaceBalearesPVP	4	FotovoltaicaPVP	4	InterconexionesPVP	8
FotovoltaicaPVP	47.5	CicloCombinadoPVP	4	VientoMadrid	4	PresionMadrid	8
HumedadMadrid	47.5	FotovoltaicaPVP	4	PresionMadrid	4	HumedadGibraltar	8
VientoGibraltar	43.75	HidraulicaPVP	4	HumedadGibraltar	4	PresionGibraltar	8
HidraulicaPVP	42.5	RegSecundariaSubir	4	día	4	DemandaPVP	7
InterconexionesPVP	42.5	HumedadMadrid	4	CogeneracionPVP	3	EolicaPVP	7
EnlaceBalearesPVP	41.25	PresionMadrid	4	HidraulicaPVP	3	HidraulicaPVP	7
CarbonPVP	40	HumedadGibraltar	4	InterconexionesPVP	3	mes	7
RegSecundariaSubir	40	mes	4	TermosolarPVP	3	BombeoPVP	6
TemperaturaVigo	38.75	PrecioMercadoDiario	4	TurbinacionBombeoPVP	3	CogeneracionPVP	6
CogeneracionPVP	36.25	CogeneracionPVP	3	VientoGibraltar	3	RegSecundariaSubir	6
CicloCombinadoPVP	35	DemandaPVP	3	mes	3	TermosolarPVP	6
HumedadVigo	33.75	ReservaPotencia	3	RegSecundariaSubir	2	TurbinacionBombeoPVP	6
DemandaPVP	31.25	TermosolarPVP	3	TemperaturaGibraltar	2	TemperaturaMadrid	6
TermosolarPVP	30	TurbinacionBombeoPVP	3	TemperaturaVigo	2	PrecioMercadoDiario	6
PrecioMercadoDiario	28.75	PresionGibraltar	3	HumedadVigo	2	VientoMadrid	5
EolicaPVP	27.5	BombeoPVP	2	PrecioMercadoDiario	2	EnlaceBalearesPVP	4
RegSecundariaBajar	22.5	RegSecundariaBajar	2	RegSecundariaBajar	1	HumedadMadrid	4
BombeoPVP	21.25	TemperaturaVigo	2	VientoVigo	1	TemperaturaVigo	4
VientoMadrid	21.25	HumedadVigo	2	EnlaceBalearesPVP	0	HumedadVigo	4
VientoVigo	21.25	EolicaPVP	1	ReservaPotencia	0	ReservaPotencia	3
TurbinacionBombeoPVP	17.5	VientoMadrid	1	TemperaturaMadrid	0	RegSecundariaBajar	3
ReservaPotencia	7.5	VientoVigo	1	HumedadMadrid	0	VientoVigo	2
PresionVigo	0	PresionVigo	0	PresionVigo	0	PresionVigo	0

Tabla 3.2 - Ranking de selecciones realizadas

Poniendo atención en las distintas selecciones y en el ranking final, se observan casos curiosos. Por un lado, en las tres selecciones realizadas, se considera que el resultado del mercado diario de la nuclear (*NuclearPVP*) es un buen atributo para predecir desvíos. De hecho, es el atributo que lidera el ranking al realizar la media ponderada. Por otro lado, el resultado de la eólica (*EolicaPVP*) resulta importante bajo el punto de vista de los árboles de decisión pero no por el contrario para las redes neuronales, con lo que este atributo cae a la mitad de la tabla en la

clasificación general. Algo parecido pero al revés ocurre para *TemperaturaMadrid*, el cual está muy bien considerado por las redes neuronales pero resulta ser uno de los últimos para los árboles de decisión. Para otros atributos como *PresionVigo*, ninguno de los métodos presenta dudas en posicionarlo todas las ocasiones en el último lugar de la tabla.

En definitiva, para este primer estudio se ha visto que, dada la complejidad de la predicción, el programa no realiza malas clasificaciones, siempre que se parta de un grupo de atributos bien relacionados con la clase y de una variada y gran cantidad de datos de entrenamiento. Esta idea se ve reflejada en el momento en que no es posible obtener un modelo de predicción con los datos de un año para aplicarlo al siguiente, pero sí es posible obtener una predicción aceptable con los mismos datos pero de forma desordenada y aplicando el mismo porcentaje 50% entrenamiento y 50% testeo. Y a medida que se aumenta este porcentaje o se realizan validaciones cruzadas, el número de aciertos se incrementa. Por ello, se entiende que los datos de entrenamiento para la realización de un modelo de predicción, deben encontrarse lo más próximo posible en el tiempo a los datos que se quieren predecir. Del estudio también se extrae la idea de que no solo se debe pensar con lógica a la hora de elegir los atributos para realizar la predicción, sino que hay que tener en cuenta que pueden existir variables que pasen completamente desapercibidas pero que realmente aporten información valiosa. Por ejemplo, esto se da con los resultados del mercado diario para la generación nuclear. Por algún motivo desconocido, ya sea directo o indirecto, esta variable es capaz de discriminar mejor que otras si los desvíos serán en un sentido u otro. Por ello, en un primer momento parece recomendable incluir cuantos más atributos mejor y luego ir filtrando si es necesario.

4. HIPÓTESIS Y RESULTADOS DEL ESTUDIO FINAL

4.1. VENTANA MÓVIL

Dados los resultados del estudio anterior, se decide que lo más conveniente es utilizar datos de entrenamiento que se encuentren cercanos en el tiempo a los datos que se quieren clasificar para obtener una predicción. Por ello se plantea crear una ventana móvil de *n* días, en la cual se utilicen los *n-1* primeros días para entrenar el modelo y el último para testear. Se utilizará esta ventana móvil para predecir el sentido de los desvíos del sistema para los 59 días comprendidos entre el 1 de Enero de 2018 y el 28 de Febrero de 2018. Se utilizarán los dos métodos estudiados hasta ahora (árboles de decisión y redes neuronales), con distintas configuraciones de sus parámetros y con distintas dimensiones para la ventana móvil.

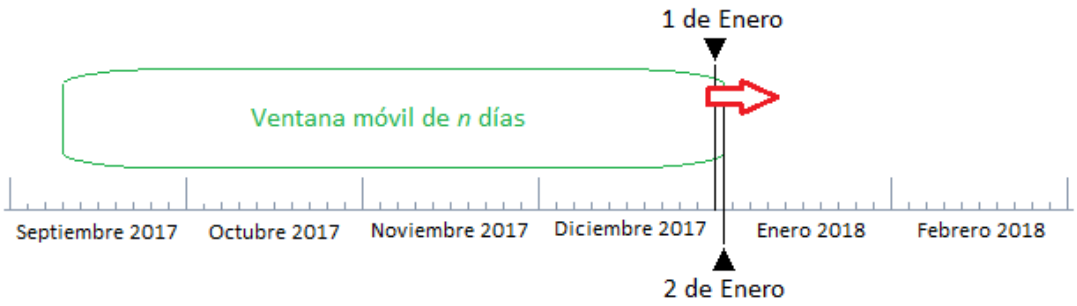


Figura 4.1 - Ventana móvil de entrenamiento y testeo

Para la primera ventana móvil se creará una primera muestra de 100 días y si se pretende que el último día de esta ventana sea el 1 de Enero de 2018, el primer día de la ventana deberá ser el 24 de Septiembre de 2017. Se escoge la muestra de 100 días porque permite realizar de un modo sencillo la división entrenamiento-testeo que se quiere hacer, correspondiendo un 99% a entrenamiento y el 1% restante a testeo. Una vez se obtengan los resultados, se puede ampliar o reducir esta ventana para comprobar si los resultados mejoran o empeoran en función de su longitud. Por lo tanto, ya se tiene definida una longitud para la ventana y un periodo temporal por el que se va a deslizar.

Lo próximo, será crear los ficheros de entrada al programa con las distintas versiones de la ventana móvil, es decir, las 59 muestras en las que se va sumando un día en la cabeza y se le va restando uno en la cola. Esto es sencillo, ya que si en un principio se creó una función de Matlab que creaba ficheros de entrada a WEKA con datos que cumpliesen una serie de características, ahora solo se debe modificar tal función para que, tras definir la ventana requerida, entre en un bucle y vaya creando las 59 muestras de 2400 horas cada una. Una vez que se tienen las 59 muestras, no será necesario ir al *Explorer* y simularlas una a una. Gracias a que se cuenta con la

herramienta del *Experimenter*, se pueden introducir como entrada las 59 muestras, elegir varios clasificadores distintos, marcar las condiciones de clasificación (muestra ordenada con 99% para entrenamiento y 1% para testeo) y comenzar a simular. Estas simulaciones pueden llevar bastante tiempo en realizarse, ya que son 59 muestras y según los métodos de clasificación escogidos, los tiempos de computación pueden ser largos. Pero esta herramienta presenta la comodidad de poder establecer parámetros para múltiples experimentos en el inicio y obtener al final, resultados que orienten en gran medida hacia donde debe ir el estudio, como conocer qué longitud es la óptima para la ventana o que método aporta mejores resultados.

Lo siguiente será definir los métodos de clasificación que se utilizarán para testear las 59 muestras de la ventana. A continuación se enumeran los métodos elegidos para hacer una primera aproximación:

1. Clasificador *ZeroR.*

2. Clasificador *J48* con 0.01 de *confidenceFactor*.

3. Clasificador *J48* con 0.25 de *confidenceFactor*.

4. Clasificador *J48* con 0.01 de *confidenceFactor* tras haber realizado una selección de atributos con Metaclasificador *AttributeSelectedClasifier* con método *wrapper* que use *J48* como clasificador base.

5. Clasificador *J48* con 0.25 de *confidenceFactor* tras haber realizado una selección de atributos con Metaclasificador *AttributeSelectedClasifier* con método *wrapper* que use *J48* como clasificador base.

6. Clasificador MultilayerPerceptron con 0.3 en *learningRate*, 0.2 en *momentum* y "*a*" neuronas en una capa oculta.

7. Clasificador MultilayerPerceptron con 0.6 en *learningRate*, 0.2 en *momentum* y "*t*" neuronas en una capa oculta.

8. Clasificador MultilayerPerceptron con 0.4 en *learningRate*, 0.1 en *momentum* y "*t*" neuronas en una capa oculta.

9. Clasificador MultilayerPerceptron con 0.3 en *learningRate*, 0.1 en *momentum* y "*a*" neuronas en una capa oculta tras haber realizado una selección de atributos con Metaclasificador *AttributeSelectedClasifier* con método *GreedyStepwise* que use *MultilayerPerceptron* como clasificador base.

Así, se tendrán resultados para ambos clasificadores principales (árboles de decisión y redes neuronales), con distintas configuraciones de parámetros más y menos complejas y siempre comparando frente a los resultados de *ZeroR*, el cual es el clasificador a superar. En la siguiente imagen (*figura 4.2*), pueden verse los resultados obtenidos para la ventana móvil de 100 días y utilizando las 9 configuraciones anteriormente descritas. En cada una de las columnas se encuentran los resultados para cada uno de los métodos y es posible identificar cual es cada uno debido a que se han numerado del mismo modo en que se han descrito.

Dataset		(1) rules.Zer	(2) trees.	(3) trees.	(4) meta.	(5) meta.	(6) functi	(7) functi	(8) functi	(9) meta.
ESTUDIOFINAL01	(1)	0.00	58.33 v	58.33 v	33.33 v	29.17 v	0.00	58.33 v	100.00 v	4.17 v
ESTUDIOFINAL02	(1)	0.00	95.83 v	95.83 v	33.33 v	12.50 v	100.00 v	100.00 v	100.00 v	50.00 v
ESTUDIOFINAL03	(1)	8.33	91.67 v	91.67 v	62.50 v	87.50 v	91.67 v	91.67 v	91.67 v	91.67 v
ESTUDIOFINAL04	(1)	12.50	87.50 v	87.50 v	70.83 v	70.83 v	87.50 v	87.50 v	87.50 v	87.50 v
ESTUDIOFINAL05	(1)	0.00	100.00 v	100.00 v	0.00	0.00	100.00 v	100.00 v	100.00 v	100.00 v
ESTUDIOFINAL06	(1)	95.83	4.17 *	4.17 *	95.83	79.17 *	20.83 *	8.33 *	4.17 *	4.17 *
ESTUDIOFINAL07	(1)	79.17	79.17	79.17	79.17	83.33 v	79.17	87.50 v	79.17	79.17
ESTUDIOFINAL08	(1)	70.83	29.17 *	29.17 *	50.00 *	50.00 *	66.67 *	33.33 *	75.00 v	41.67 *
ESTUDIOFINAL09	(1)	91.67	58.33 *	58.33 *	33.33 *	33.33 *	91.67	54.17 *	83.33 *	41.67 *
ESTUDIOFINAL10	(1)	54.17	58.33 v	54.17	54.17	54.17	58.33 v	66.67 v	58.33 v	58.33 v
ESTUDIOFINAL11	(1)	58.33	66.67 v	58.33	16.67 *	16.67 *	54.17 *	37.50 *	62.50 v	25.00 *
⋮		⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
ESTUDIOFINAL47	(1)	50.00	45.83 *	83.33 v	20.83 *	20.83 *	66.67 v	75.00 v	70.83 v	50.00
ESTUDIOFINAL48	(1)	25.00	66.67 v	83.33 v	41.67 v	66.67 v	50.00 v	79.17 v	50.00 v	70.83 v
ESTUDIOFINAL49	(1)	0.00	50.00 v	58.33 v	33.33 v	33.33 v	79.17 v	66.67 v	66.67 v	66.67 v
ESTUDIOFINAL50	(1)	25.00	54.17 v	62.50 v	54.17 v	54.17 v	70.83 v	75.00 v	79.17 v	66.67 v
ESTUDIOFINAL51	(1)	20.83	58.33 v	58.33 v	79.17 v	79.17 v	79.17 v	87.50 v	83.33 v	83.33 v
ESTUDIOFINAL52	(1)	37.50	58.33 v	58.33 v	66.67 v	62.50 v	50.00 v	58.33 v	66.67 v	62.50 v
ESTUDIOFINAL53	(1)	33.33	83.33 v	62.50 v	66.67 v	66.67 v	83.33 v	66.67 v	75.00 v	58.33 v
ESTUDIOFINAL54	(1)	54.17	75.00 v	62.50 v	45.83 *	45.83 *	66.67 v	58.33 v	75.00 v	87.50 v
ESTUDIOFINAL55	(1)	54.17	66.67 v	66.67 v	87.50 v	87.50 v	79.17 v	87.50 v	87.50 v	79.17 v
ESTUDIOFINAL56	(1)	41.67	87.50 v	83.33 v	70.83 v	66.67 v	75.00 v	83.33 v	79.17 v	83.33 v
ESTUDIOFINAL57	(1)	58.33	62.50 v	58.33	66.67 v	66.67 v	62.50 v	58.33	50.00 *	54.17 *
ESTUDIOFINAL58	(1)	87.50	79.17 *	79.17 *	83.33 *	83.33 *	87.50	83.33 *	87.50	87.50
ESTUDIOFINAL59	(1)	100.00	100.00	100.00	58.33 *	79.17 *	100.00	91.67 *	100.00	62.50 *
Average		50.78	63.35	62.57	57.63	57.42	67.16	67.73	70.62	62.57
	(v/ /*)		(37/3/19)	(32/10/17	(32/11/16)	(34/6/19)	(34/9/16)	(37/5/17)	(37/8/14)	(35/6/18)

Figura 4.2 - Resultados para ventana móvil de 100 días

Cada columna corresponde a cada uno de los métodos enumerados y cada fila a cada una de las 59 versiones de la ventana móvil, con lo que es posible ver fácilmente que resultado ha arrojado cada método con cada muestra de datos. Al final de la tabla se puede ver una media de los resultados y una comparación de cada método con el método base (*ZeroR*). En esta comparación se pueden ver tres valores entre paréntesis y separados por barras. Estos valores se rigen por la codificación que se puede ver a la izquierda de la fila (v/ /*). Los números en la posición de la "*v*", son las veces que el método en cuestión ha superado al método base. Los que se encuentren en la posición del "*", serán las veces que el método ha arrojado resultados inferiores a los del método base. Y por último, los números que se encuentren en la posición donde no hay símbolo, serán las veces en que el método ha arrojado un resultado estadísticamente similar al del método base. Se puede comprobar como el método base, de media arroja resultados del 50% de aciertos. En ciertas ocasiones obtiene 100% de aciertos debido a que en el día en cuestión todos los desvíos fueron en el sentido mayoritario pero en muchas otras ocasiones también ofrece 0% de aciertos con lo que se ve la irregularidad e ineficacia del método. Donde se han obtenido mejores resultados es utilizando el octavo método de la lista. Este corresponde al perceptrón multicapa con 0.4 de *learningRate*, 0.1 de *momentum* y "*t*" neuronas en una capa oculta, lo cual correspondía al número de atributos más la clase. Con un 70.62% de aciertos de media, supera en 20 puntos los resultados del clasificador base. Esta media podría ser algo superior, pero debido a algunos malos resultados puntuales, esta se ve disminuida.

A continuación, se realizará la misma simulación para el periodo de tiempo dado y con ventanas móviles de 50 y 200 días. Lo único que cambiará en este caso es la magnitud de las 59 muestras de entrada y el porcentaje de entrenamiento y testeo, el cual será de 98% para la ventana de 50

días y de 99.5% para la de 200 días. Los resultados pueden verse en las siguientes imágenes (*figura 4.3* y *figura 4.4*).

Dataset		(1) rules.Zer	(2) trees.	(3) trees.	(4) meta.	(5) meta.	(6) functi	(7) functi	(8) functi	(9) meta.A
ESTUDIOFINAL1	(1)	0.00	54.17 v	54.17 v	37.50 v	37.50 v	0.00	0.00	0.00	37.50 v
ESTUDIOFINAL2	(1)	0.00	87.50 v	87.50 v	0.00	33.33 v	100.00 v	100.00 v	100.00 v	100.00 v
ESTUDIOFINAL3	(1)	8.33	91.67 v	91.67 v	91.67 v	91.67 v	91.67 v	91.67 v	91.67 v	91.67 v
ESTUDIOFINAL4	(1)	12.50	87.50 v	87.50 v	54.17 v	54.17 v	87.50 v	87.50 v	87.50 v	83.33 v
ESTUDIOFINAL5	(1)	0.00	100.00 v	100.00 v	95.83 v	95.83 v	87.50 v	95.83 v	87.50 v	91.67 v
ESTUDIOFINAL6	(1)	95.83	4.17 *	4.17 *	45.83 *	41.67 *	16.67 *	4.17 *	4.17 *	4.17 *
ESTUDIOFINAL7	(1)	79.17	79.17	79.17	41.67 *	41.67 *	75.00 *	75.00 *	45.83 *	79.17
ESTUDIOFINAL8	(1)	70.83	29.17 *	29.17 *	37.50 *	37.50 *	29.17 *	41.67 *	66.67 *	33.33 *
ESTUDIOFINAL9	(1)	91.67	58.33 *	58.33 *	91.67	91.67	87.50 *	66.67 *	70.83 *	37.50 *
ESTUDIOFINAL10	(1)	54.17	45.83 *	45.83 *	58.33 v	58.33 v	41.67 *	50.00 *	37.50 *	58.33 v
ESTUDIOFINAL11	(1)	58.33	70.83 v	37.50 *	20.83 *	20.83 *	41.67 *	37.50 *	33.33 *	25.00 *
⋮		⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
ESTUDIOFINAL47	(1)	50.00	58.33 v	54.17 v	95.83 v	95.83 v	41.67 *	50.00	58.33 v	54.17 v
ESTUDIOFINAL48	(1)	25.00	75.00 v	75.00 v	70.83 v	62.50 v	58.33 v	70.83 v	62.50 v	66.67 v
ESTUDIOFINAL49	(1)	0.00	54.17 v	45.83 v	0.00	0.00	70.83 v	83.33 v	95.83 v	20.83 v
ESTUDIOFINAL50	(1)	25.00	62.50 v	62.50 v	50.00 v	50.00 v	75.00 v	79.17 v	79.17 v	75.00 v
ESTUDIOFINAL51	(1)	20.83	75.00 v	75.00 v	62.50 v	62.50 v	79.17 v	79.17 v	91.67 v	70.83 v
ESTUDIOFINAL52	(1)	37.50	70.83 v	50.00 v	37.50	33.33 *	58.33 v	66.67 v	58.33 v	66.67 v
ESTUDIOFINAL53	(1)	33.33	62.50 v	62.50 v	75.00 v	75.00 v	66.67 v	75.00 v	58.33 v	70.83 v
ESTUDIOFINAL54	(1)	54.17	75.00 v	66.67 v	66.67 v	66.67 v	79.17 v	66.67 v	66.67 v	75.00 v
ESTUDIOFINAL55	(1)	54.17	91.67 v	91.67 v	87.50 v	87.50 v	79.17 v	91.67 v	79.17 v	83.33 v
ESTUDIOFINAL56	(1)	41.67	54.17 v	83.33 v	58.33 v	58.33 v	83.33 v	87.50 v	83.33 v	75.00 v
ESTUDIOFINAL57	(1)	58.33	54.17 *	45.83 *	45.83 *	50.00 *	66.67 v	54.17 *	54.17 *	33.33 *
ESTUDIOFINAL58	(1)	87.50	87.50	87.50	87.50	87.50	50.00 *	79.17 *	50.00 *	41.67 *
ESTUDIOFINAL59	(1)	100.00	62.50 *	66.67 *	50.00 *	91.67 *	100.00	95.83 *	79.17 *	100.00
Average		43.57	64.48	62.78	54.45	55.08	65.61	66.74	64.69	62.22
		(v/ /*)	(40/6/13)	(39/5/15	(32/13/14	(33/11/15)	(41/4/14)	(39/4/16)	(39/6/14)	(37/7/15)

Figura 4.3 - Resultados para ventana móvil de 50 días

Dataset		(1) rules.Zer	(2) trees.	(3) trees.	(4) meta.A	(5) meta.A	(6) functi	(7) functi	(8) functi	(9) meta.A
ESTUDIOFINAL01	(1)	0.00	66.67 v	66.67 v	12.50 v	12.50 v	70.83 v	100.00 v	100.00 v	54.17 v
ESTUDIOFINAL02	(1)	0.00	79.17 v	79.17 v	91.67 v	0.00	95.83 v	100.00 v	100.00 v	100.00 v
ESTUDIOFINAL03	(1)	8.33	91.67 v	91.67 v	91.67 v	91.67 v	91.67 v	91.67 v	91.67 v	91.67 v
ESTUDIOFINAL04	(1)	12.50	87.50 v	87.50 v	87.50 v	87.50 v	87.50 v	87.50 v	87.50 v	87.50 v
ESTUDIOFINAL05	(1)	100.00	100.00	100.00	100.00	100.00	100.00	91.67 *	95.83 *	100.00
ESTUDIOFINAL06	(1)	4.17	4.17	4.17	4.17	8.33 v	4.17	4.17	4.17	4.17
ESTUDIOFINAL07	(1)	20.83	79.17 v	79.17 v	79.17 v	79.17 v	33.33 v	79.17 v	75.00 v	62.50 v
ESTUDIOFINAL08	(1)	29.17	29.17	29.17	45.83 v	16.67 *	29.17	29.17	25.00 *	33.33 v
ESTUDIOFINAL09	(1)	8.33	58.33 v	58.33 v	25.00 v	25.00 v	66.67 v	54.17 v	20.83 v	29.17 v
ESTUDIOFINAL10	(1)	45.83	37.50 *	37.50 *	54.17 v	54.17 v	58.33 v	54.17 v	45.83	54.17 v
ESTUDIOFINAL11	(1)	41.67	20.83 *	20.83 *	41.67	41.67	58.33 v	50.00 v	62.50 v	58.33 v
⋮		⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
ESTUDIOFINAL47	(1)	50.00	45.83 *	66.67 v	75.00 v	50.00	70.83 v	58.33 v	62.50 v	58.33 v
ESTUDIOFINAL48	(1)	25.00	66.67 v	62.50 v	62.50 v	66.67 v	75.00 v	37.50 v	62.50 v	83.33 v
ESTUDIOFINAL49	(1)	0.00	54.17 v	62.50 v	54.17 v	54.17 v	58.33 v	62.50 v	58.33 v	62.50 v
ESTUDIOFINAL50	(1)	25.00	66.67 v	41.67 v	66.67 v	54.17 v	41.67 v	58.33 v	62.50 v	54.17 v
ESTUDIOFINAL51	(1)	20.83	83.33 v	83.33 v	66.67 v	50.00 v	79.17 v	70.83 v	66.67 v	58.33 v
ESTUDIOFINAL52	(1)	37.50	54.17 v	58.33 v	66.67 v	58.33 v	50.00 v	58.33 v	70.83 v	58.33 v
ESTUDIOFINAL53	(1)	33.33	70.83 v	58.33 v	54.17 v	45.83 v	83.33 v	83.33 v	58.33 v	83.33 v
ESTUDIOFINAL54	(1)	54.17	58.33 v	58.33 v	75.00 v	75.00 v	75.00 v	66.67 v	79.17 v	79.17 v
ESTUDIOFINAL55	(1)	54.17	87.50 v	87.50 v	75.00 v	79.17 v	87.50 v	87.50 v	79.17 v	91.67 v
ESTUDIOFINAL56	(1)	41.67	75.00 v	66.67 v	54.17 v	58.33 v	79.17 v	91.67 v	75.00 v	75.00 v
ESTUDIOFINAL57	(1)	58.33	58.33	50.00 *	41.67 *	37.50 *	58.33	45.83 *	50.00 *	62.50 v
ESTUDIOFINAL58	(1)	87.50	83.33 *	83.33 *	87.50	87.50	79.17 *	87.50	75.00 *	83.33 *
ESTUDIOFINAL59	(1)	100.00	100.00	87.50 *	45.83 *	41.67 *	33.33 *	79.17 *	87.50 *	100.00
Average		48.94	63.14	61.02	59.89	56.50	65.40	65.68	64.97	62.01
		(v/ /*)	(32/8/19)	(31/7/21)	(33/10/16)	(31/12/16)	(40/7/12)	(36/6/17)	(33/7/19)	(34/6/19)

Figura 4.4 - Resultados para ventana móvil de 200 días

Se puede comprobar que para ambos casos se consiguen resultados peores que utilizando la ventana de 100 días. Si se miran los porcentajes de aciertos de los clasificadores base (43.57% y 48.94%), se puede ver que son menores que en el caso anterior (50.78%). Esto hace que en ambas simulaciones, la diferencia entre el porcentaje de la mejor clasificación y el porcentaje del clasificador base sean mayores. Es decir, que se ha superado en mayor medida lo que se podía considerar el mínimo de aciertos. Pero a efectos prácticos, la predicción resultante es peor, por lo tanto la ventana que interesa utilizar es la de 100 días. También se ve, que dentro de las distintas alternativas elegidas como método de predicción, la que mejor resultados ha dado es la octava de la lista, correspondiente al *MultilayerPerceptron (L-0.4 M-0.1 H "t")*. Esta última forma de representar el clasificador y sus parámetros será una manera común de verlo dentro del programa, refiriéndose al perceptrón multicapa con el valor de los parámetros típicos ya descritos.

También es interesante, una vez se ha acotado un método y unos parámetros de configuración, realizar pruebas con distintas configuraciones cercanas a esta para ver si se está en el óptimo o se puede mejorar la predicción en cierta medida. Por tanto, se volverá a simular incluyendo las siguientes configuraciones:

1. Clasificador MultilayerPerceptron con 0.4 en *learningRate*, 0.1 en *momentum* y "*t*" neuronas en una capa oculta.

2. Clasificador *ZeroR.*

3. Clasificador MultilayerPerceptron con 0.5 en *learningRate*, 0.1 en *momentum* y "*t*" neuronas en una capa oculta.

4. Clasificador MultilayerPerceptron con 0.6 en *learningRate*, 0.1 en *momentum* y "*t*" neuronas en una capa oculta.

5. Clasificador MultilayerPerceptron con 0.7 en *learningRate*, 0.1 en *momentum* y "*t*" neuronas en una capa oculta.

6. Clasificador MultilayerPerceptron con 0.7 en *learningRate*, 0.2 en *momentum* y "*t*" neuronas en una capa oculta.

Se coloca en esta ocasión en la primera columna, el clasificador que obtuvo mejores resultados en la anterior simulación para la ventana móvil de 100 días. Se sigue incluyendo el clasificador *ZeroR*, pero esta vez interesa comparar el resto de métodos con el que hasta ahora se cree el más fiable. El resultado puede verse en la siguiente imagen (*figura 4.5*).

Dataset	(1) functions	(2) rules.	(3) functi	(4) functi	(5) functi	(6) functi
ESTUDIOFINAL01	(1) 100.00	0.00 *	100.00	100.00	33.33 *	100.00
ESTUDIOFINAL02	(1) 100.00	0.00 *	100.00	100.00	100.00	100.00
ESTUDIOFINAL03	(1) 91.67	8.33 *	91.67	91.67	91.67	91.67
ESTUDIOFINAL04	(1) 87.50	12.50 *	87.50	87.50	87.50	87.50
ESTUDIOFINAL05	(1) 100.00	0.00 *	100.00	100.00	100.00	100.00
ESTUDIOFINAL06	(1) 4.17	95.83 v	4.17	4.17	4.17	8.33 v
ESTUDIOFINAL07	(1) 79.17	79.17	79.17	83.33 v	79.17	75.00 *
ESTUDIOFINAL08	(1) 75.00	70.83 *	20.83 *	54.17 *	54.17 *	29.17 *
ESTUDIOFINAL09	(1) 83.33	91.67 v	58.33 *	41.67 *	83.33	62.50 *
ESTUDIOFINAL10	(1) 58.33	54.17 *	62.50 v	50.00 *	62.50 v	54.17 *
ESTUDIOFINAL11	(1) 62.50	58.33 *	70.83 v	45.83 *	87.50 v	62.50
⋮	⋮	⋮	⋮	⋮	⋮	⋮
ESTUDIOFINAL47	(1) 70.83	50.00 *	50.00 *	66.67 *	66.67 *	58.33 *
ESTUDIOFINAL48	(1) 50.00	25.00 *	66.67 v	50.00	58.33 v	54.17 v
ESTUDIOFINAL49	(1) 66.67	0.00 *	70.83 v	70.83 v	58.33 *	75.00 v
ESTUDIOFINAL50	(1) 79.17	25.00 *	79.17	66.67 *	70.83 *	66.67 *
ESTUDIOFINAL51	(1) 83.33	20.83 *	79.17 *	87.50 v	70.83 *	79.17 *
ESTUDIOFINAL52	(1) 66.67	37.50 *	75.00 v	54.17 *	58.33 *	50.00 *
ESTUDIOFINAL53	(1) 75.00	33.33 *	75.00	58.33 *	83.33 v	75.00
ESTUDIOFINAL54	(1) 75.00	54.17 *	70.83 *	66.67 *	75.00	58.33 *
ESTUDIOFINAL55	(1) 87.50	54.17 *	87.50	79.17 *	79.17 *	79.17 *
ESTUDIOFINAL56	(1) 79.17	41.67 *	83.33 v	87.50 v	79.17	70.83 *
ESTUDIOFINAL57	(1) 50.00	58.33 v	54.17 v	62.50 v	50.00	50.00
ESTUDIOFINAL58	(1) 87.50	87.50	54.17 *	83.33 *	87.50	79.17 *
ESTUDIOFINAL59	(1) 100.00	100.00	100.00	100.00	100.00	100.00
Average	70.62	50.78	68.29	67.87	66.45	65.25
	(v/ /*)	(14/8/37)	(23/20/16)	(16/16/27)	(16/14/29)	(12/15/32)

Figura 4.5 - Segundos resultados para ventana móvil de 100 días

Una vez vistos los resultados, resulta curioso que la configuración de la tercera columna, sea considerada mejor estadísticamente que la configuración de base, aun obteniendo esta última un mejor porcentaje de aciertos de media. Es decir, la configuración de la tercera columna, ha resultado ser estadísticamente mejor que la de base en 23 ocasiones y peor en 16. Esto es debido a que las veces que haya sido mejor, habrá obtenido un porcentaje no muy superior y por el contrario, las veces que haya sido peor, habrá obtenido un porcentaje algo más inferior. Por lo tanto, parece prudente seguir confiando en la primera de las configuraciones para el clasificador *MultilayerPerceptron* ya que a efectos prácticos, ha sido la que más horas ha acertado en la predicción total. El porcentaje de aciertos de 70.62% equivale a prácticamente 17 horas acertadas por cada 24.

Una vez llegado a este punto, en el que se ha definido el método de predicción que parece ajustarse mejor a la muestra, se deberá hacer un análisis de las predicciones realizadas. La herramienta del *Experimenter* permite obtener de manera cómoda unos resultados que orientan a la hora de escoger un clasificador, unos parámetros de configuración y unas muestras de entrenamiento, pero tiene el inconveniente de que no genera un fichero con las predicciones

que se han hecho para cada simulación. Esto en cambio sí podía hacerse desde el *Explorer*, simulando una a una cada muestra. De querer hacer esto con los 9 métodos anteriores, sería algo complicado, debido a la cantidad de simulaciones a realizar y predicciones que exportar, con lo que el *Experimenter* ayuda a acotar esta tarea. Una vez recogidas las predicciones realizadas para cada uno de los 59 días con el método elegido, se agrupan los resultados obtenidos por horas y se desglosan en aciertos y fallos para cada uno de los sentidos de desvíos como se muestra en la siguiente tabla (*tabla 4.1*).

Hora del día	Desvíos a subir	Desvíos a bajar	Aciertos	Desvíos a subir acertados	Desvíos a bajar acertados	Fallos	Desvíos a bajar fallados	Desvíos a subir fallados
1	35	24	42	24	18	17	6	11
2	26	33	42	20	22	17	11	6
3	19	40	42	12	30	17	10	7
4	18	41	39	8	31	20	10	10
5	13	46	48	9	39	11	7	4
6	12	47	46	7	39	13	8	5
7	17	42	41	9	32	18	10	8
8	21	38	37	10	27	22	11	11
9	27	32	38	14	24	21	8	13
10	26	33	38	14	24	21	9	12
11	23	36	39	15	24	20	12	8
12	27	32	40	21	19	19	13	6
13	28	31	39	23	16	20	15	5
14	34	25	43	30	13	16	12	4
15	41	18	46	36	10	13	8	5
16	43	16	48	39	9	11	7	4
17	38	21	46	34	12	13	9	4
18	34	25	42	30	12	17	13	4
19	35	24	44	27	17	15	7	8
20	29	30	44	26	18	15	12	3
21	33	26	38	27	11	21	15	6
22	40	19	36	30	6	23	13	10
23	40	19	38	31	7	21	12	9
24	38	21	44	32	12	15	9	6

Tabla 4.1 - Resultado de las predicciones realizadas para la ventana de 100 días

Agrupar los resultados de este modo, posibilita saber cuáles son las horas donde más aciertos y fallos se están cometiendo. También permite ver la naturaleza de estos aciertos y estos fallos, por lo que en definitiva, se está generando una matriz de confusión para cada hora del día. Esto resulta bastante útil, ya que permitirá añadir penalizaciones en la matriz de costes en determinadas horas según se crea conveniente. También sería interesante añadir una columna más con los costes medios de desviarse a bajar o a subir, con lo que se tendría una información más completa para tomar una decisión sobre como alterar la matriz de costes según la hora que se esté prediciendo. En la siguiente imagen (*figura 4.6*), se puede ver la representación de los fallos y aciertos junto con el número de desvíos a subir y a bajar que se han dado.

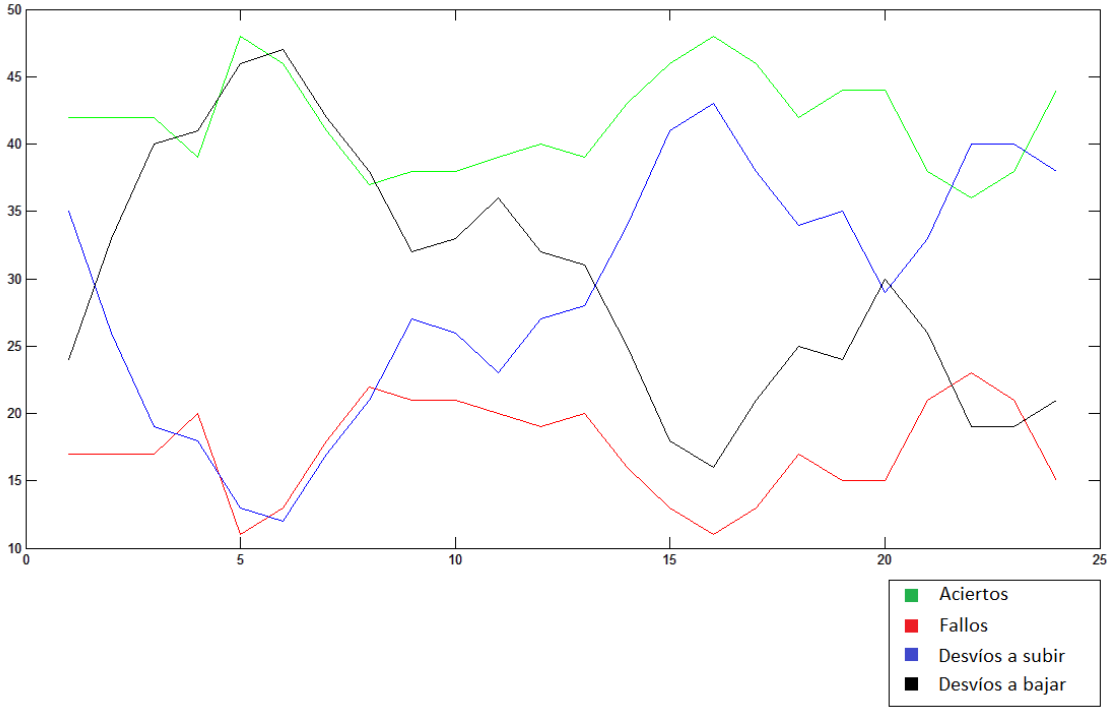


Figura 4.6 - Representación gráfica de fallos y aciertos totales junto al número y sentido de los desvíos

Algo que ya se sabía, extraído del proyecto anterior, es que el mayor número de desvíos a bajar se concentraban en las horas de la madrugada y el mayor número de desvíos a subir, en las horas de punta de demanda, tal como se muestra en la imagen. Del mismo modo, el mayor número de aciertos también se concentran en estos dos momentos. Es algo beneficioso que así sea, ya que los desvíos a bajar, son generalmente más caros en las horas de madrugada, y los desvíos a subir son más caros en las horas de punta de demanda, tal como puede verse en el ejemplo de la imagen siguiente (*figura 4.7*), extraída del anterior trabajo.

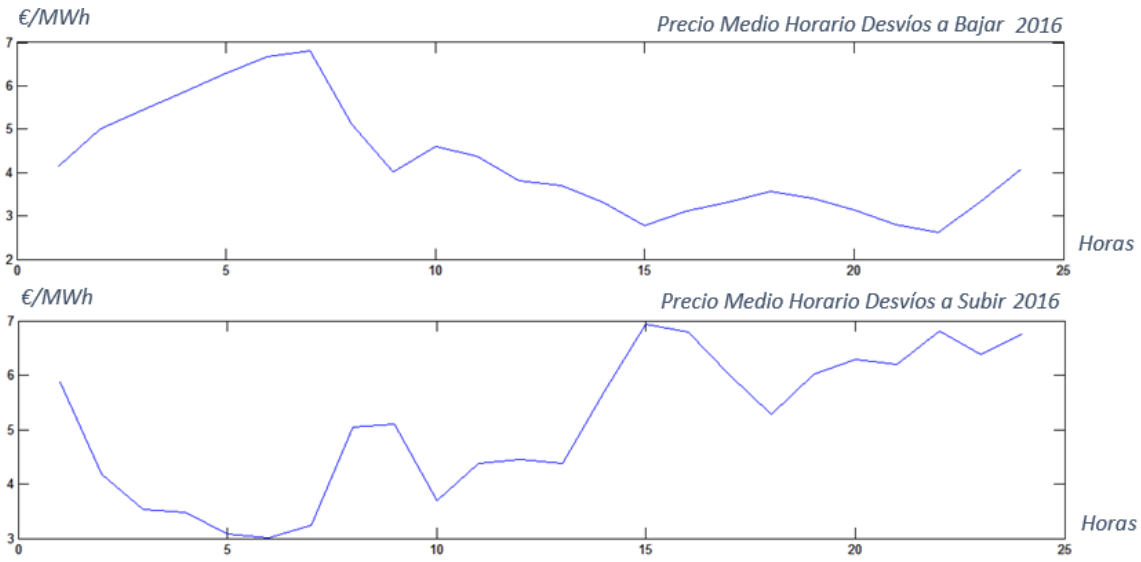


Figura 4.7 - Precio medio horario de los desvíos para el año 2016

En la siguiente tabla (*tabla 4.2*), se muestran las 24 matrices de confusión generadas a raíz de la agrupación de aciertos y fallos de las predicciones de los 59 días.

Horas	Matriz de confusión	Horas	Matriz de confusión	Horas	Matriz de confusión
1	$\begin{bmatrix}24 & 11\\ 6 & 18\end{bmatrix}$	9	$\begin{bmatrix}14 & 13\\ 8 & 24\end{bmatrix}$	17	$\begin{bmatrix}34 & 4\\ 9 & 12\end{bmatrix}$
2	$\begin{bmatrix}20 & 6\\ 11 & 22\end{bmatrix}$	10	$\begin{bmatrix}14 & 12\\ 9 & 24\end{bmatrix}$	18	$\begin{bmatrix}30 & 4\\ 13 & 12\end{bmatrix}$
3	$\begin{bmatrix}12 & 7\\ 10 & 30\end{bmatrix}$	11	$\begin{bmatrix}15 & 8\\ 12 & 24\end{bmatrix}$	19	$\begin{bmatrix}27 & 8\\ 7 & 17\end{bmatrix}$
4	$\begin{bmatrix}8 & 10\\ 10 & 31\end{bmatrix}$	12	$\begin{bmatrix}21 & 6\\ 13 & 19\end{bmatrix}$	20	$\begin{bmatrix}26 & 3\\ 12 & 18\end{bmatrix}$
5	$\begin{bmatrix}9 & 4\\ 7 & 39\end{bmatrix}$	13	$\begin{bmatrix}23 & 5\\ 15 & 16\end{bmatrix}$	21	$\begin{bmatrix}27 & 6\\ 15 & 11\end{bmatrix}$
6	$\begin{bmatrix}7 & 5\\ 8 & 39\end{bmatrix}$	14	$\begin{bmatrix}30 & 4\\ 12 & 13\end{bmatrix}$	22	$\begin{bmatrix}30 & 10\\ 13 & 6\end{bmatrix}$
7	$\begin{bmatrix}9 & 8\\ 10 & 32\end{bmatrix}$	15	$\begin{bmatrix}36 & 5\\ 8 & 10\end{bmatrix}$	23	$\begin{bmatrix}31 & 9\\ 12 & 7\end{bmatrix}$
8	$\begin{bmatrix}10 & 11\\ 11 & 27\end{bmatrix}$	16	$\begin{bmatrix}39 & 4\\ 7 & 9\end{bmatrix}$	24	$\begin{bmatrix}32 & 6\\ 9 & 12\end{bmatrix}$

Tabla 4.2 - Matrices de confusión para las 24 horas del período de 59 días

Las distintas posiciones de la matriz llevan asociadas los fallos y aciertos tal como a continuación se detallan:

- (1,1). Número de desvíos a subir acertados.
- (2,2). Número de desvíos a bajar acertados.
- (1,2). Número de horas que siendo desvíos a subir, se clasificaron erróneamente como desvíos a bajar.
- (2,1). Número de horas que siendo desvíos a bajar, se clasificaron erróneamente como desvíos a subir.

Ocupando cada posición el lugar que a continuación se indica:

$$\begin{bmatrix}(1,1) & (1,2)\\ (2,1) & (2,2)\end{bmatrix} \qquad (Ecuación\ 4)$$

Por lo tanto, creando una tabla similar a la anterior que incluya la información del precio de los desvíos para cada hora, junto con la que ya se tiene de las matrices de confusión, permitirá conocer al completo, hacia donde se están cometiendo más fallos en la predicción y si interesa

o no, incluir penalizaciones en la matriz de coste, tal como se explicó en el capítulo de metodología (*figura 2.15*). Si se hace este análisis, probablemente pueda mejorarse el casi 71% de aciertos medios que se ha obtenido y la disminución de las pérdidas por desvíos.

4.2. OTROS RESULTADOS

En el primer estudio, donde se realizaban predicciones por parejas de años, primero se obtuvieron resultados para los desvíos del sistema y posteriormente se hizo lo propio para el resto de desvíos (demanda, generación eólica y fotovoltaica). Por lo que en este caso, se hará lo mismo. Se realizará el mismo estudio de la ventana móvil, pero esta vez aplicando únicamente la ventana de 100 días. También se aplicarán los mismos métodos de clasificación exceptuando la red neuronal con 0.6 de *learningRate*, 0.2 de *momentum* y "*t*" neuronas en una capa oculta. Esto se hará debido a que, al ser el clasificador más complejo de todos, también es el que más tiempo de computación requiere y en este caso, al haber obtenido unos resultados satisfactorios para los desvíos del sistema, no se pretende hacer un estudio más profundo del que ya se ha hecho para el resto de desvíos.

Por lo tanto la lista de distintos métodos y configuraciones se presenta a continuación, reduciéndose de 9 a 8 simulaciones para cada uno de los 59 días:

1. Clasificador *ZeroR.*

2. Clasificador *J48* con 0.01 de *confidenceFactor*.

3. Clasificador *J48* con 0.25 de *confidenceFactor*.

4. Clasificador *J48* con 0.01 de *confidenceFactor* tras haber realizado una selección de atributos con Metaclasificador *AttributeSelectedClasifier* con método *wrapper* que use *J48* como clasificador base.

5. Clasificador *J48* con 0.25 de *confidenceFactor* tras haber realizado una selección de atributos con Metaclasificador *AttributeSelectedClasifier* con método *wrapper* que use *J48* como clasificador base.

6. Clasificador MultilayerPerceptron con 0.3 en *learningRate*, 0.2 en *momentum* y "*a*" neuronas en una capa oculta.

7. Clasificador MultilayerPerceptron con 0.4 en *learningRate*, 0.1 en *momentum* y "*t*" neuronas en una capa oculta.

8. Clasificador MultilayerPerceptron con 0.3 en *learningRate*, 0.1 en *momentum* y "*a*" neuronas en una capa oculta tras haber realizado una selección de atributos con Metaclasificador *AttributeSelectedClasifier* con método *GreedyStepwise* que use *MultilayerPerceptron* como clasificador base.

En la siguiente imagen (*figura 4.8*), pueden observarse los resultados para el estudio de la ventana móvil de 100 días para los desvíos de la demanda.

Dataset	(1) rules.Zer	(2) trees	(3) trees	(4) meta.A	(5) meta.A	(6) functi	(7) functi	(8) meta.
ESTUDIOFINAL1	(1) 100.00	16.67 *	41.67 *	75.00 *	75.00 *	4.17 *	0.00 *	50.00 *
ESTUDIOFINAL2	(1) 91.67	37.50 *	29.17 *	25.00 *	25.00 *	87.50 *	95.83 v	66.67 *
ESTUDIOFINAL3	(1) 91.67	91.67	79.17 *	79.17 *	91.67	79.17 *	91.67	83.33 *
ESTUDIOFINAL4	(1) 70.83	75.00 v	70.83	95.83 v	62.50 *	100.00 v	70.83	75.00 v
ESTUDIOFINAL5	(1) 87.50	87.50	87.50	87.50	87.50	87.50	91.67 v	41.67 *
ESTUDIOFINAL6	(1) 75.00	83.33 v	83.33 v	75.00	75.00	75.00	87.50 v	33.33 *
ESTUDIOFINAL7	(1) 41.67	41.67	41.67	62.50 v	70.83 v	62.50 v	62.50 v	41.67
ESTUDIOFINAL8	(1) 45.83	25.00 *	29.17 *	20.83 *	20.83 *	37.50 *	41.67 *	37.50 *
ESTUDIOFINAL9	(1) 8.33	50.00 v	70.83 v	58.33 v	58.33 v	8.33	70.83 v	29.17 v
ESTUDIOFINAL10	(1) 50.00	50.00	37.50 *	50.00	50.00	54.17 v	45.83 *	62.50 v
ESTUDIOFINAL11	(1) 41.67	62.50 v	75.00 v	41.67	37.50 *	54.17 v	45.83 v	50.00 v
⋮		⋮	⋮	⋮	⋮	⋮	⋮	⋮
ESTUDIOFINAL47	(1) 25.00	87.50 v	87.50 v	87.50 v	87.50 v	95.83 v	91.67 v	91.67 v
ESTUDIOFINAL48	(1) 29.17	66.67 v	70.83 v	33.33 v	37.50 v	79.17 v	87.50 v	83.33 v
ESTUDIOFINAL49	(1) 37.50	83.33 v	83.33 v	66.67 v	70.83 v	91.67 v	79.17 v	66.67 v
ESTUDIOFINAL50	(1) 58.33	58.33	50.00 *	58.33	58.33	70.83 v	54.17 *	62.50 v
ESTUDIOFINAL51	(1) 41.67	83.33 v	62.50 v	50.00 v	45.83 v	75.00 v	58.33 v	58.33 v
ESTUDIOFINAL52	(1) 54.17	79.17 v	79.17 v	50.00 *	50.00 *	79.17 v	75.00 v	62.50 v
ESTUDIOFINAL53	(1) 62.50	91.67 v	91.67 v	79.17 v	79.17 v	70.83 v	83.33 v	70.83 v
ESTUDIOFINAL54	(1) 79.17	70.83 *	54.17 *	66.67 *	66.67 *	70.83 *	75.00 *	79.17
ESTUDIOFINAL55	(1) 58.33	79.17 v	83.33 v	83.33 v	83.33 v	79.17 v	79.17 v	62.50 v
ESTUDIOFINAL56	(1) 50.00	91.67 v	75.00 v	50.00	50.00	95.83 v	83.33 v	87.50 v
ESTUDIOFINAL57	(1) 79.17	79.17	75.00 *	45.83 *	45.83 *	75.00 *	79.17	50.00 *
ESTUDIOFINAL58	(1) 100.00	37.50 *	54.17 *	100.00	100.00	91.67 *	95.83 *	29.17 *
ESTUDIOFINAL59	(1) 95.83	87.50 *	87.50 *	95.83	95.83	87.50 *	66.67 *	33.33 *
Average	55.58	69.77	68.57	61.58	61.09	71.54	71.75	65.47
	(v/ /*)	(37/8/14)	(35/6/18)	(32/9/18)	(30/10/19)	(38/8/13)	(40/5/14)	(38/4/17)

Figura 4.8 - Resultados para los desvíos de la demanda con ventana móvil de 100 días

Tal como era de esperar, los resultados iban a ser ligeramente mejores que los de los desvíos del sistema. También vuelve a obtener las mejores predicciones el mismo clasificador con los mismos parámetros de configuración que obtuvo los mejores resultados anteriormente. Eso sí, los resultados para los árboles de decisión mejoran notablemente en este caso, acercándose bastante a los porcentajes de aciertos medios que obtienen las redes neuronales. Además, de los árboles de decisión utilizados, el que mejor resultado obtiene (69.77%), es el que se le ha definido un *confidenceFactor* más pequeño (0.01), es decir, el árbol más sencillo de todos. Esto tiene una interpretación bastante positiva, ya que es el clasificador que más rápido genera sus modelos. No llega a tardar más de 11 segundos en realizar las 59 simulaciones. En la siguiente imagen (*figura 4.9*), se puede ver la misma tabla de resultados que se acaba de exponer, pero cambiando los porcentajes de aciertos por los tiempos de computación que ha gastado cada método en realizar el entrenamiento y generar cada uno de los modelos.

Dataset	(1)	rules.Z	(2) tree	(3) tree	(4) meta	(5) meta	(6) funct	(7) funct	(8) meta
ESTUDIOFINAL1	(1)	0.00	0.19 v	0.18 v	0.52 v	0.50 v	23.04 v	46.45 v	6.35 v
ESTUDIOFINAL2	(1)	0.00	0.18 v	0.17 v	2.67 v	2.73 v	23.18 v	47.41 v	1.11 v
ESTUDIOFINAL3	(1)	0.00	0.18 v	0.21 v	1.56 v	1.54 v	23.04 v	47.23 v	4.99 v
ESTUDIOFINAL4	(1)	0.00	0.20 v	0.22 v	0.95 v	0.96 v	23.17 v	49.17 v	3.88 v
⋮		⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
ESTUDIOFINAL55	(1)	0.00	0.17 v	0.17 v	1.43 v	1.43 v	24.01 v	46.81 v	1.69 v
ESTUDIOFINAL56	(1)	0.00	0.16 v	0.17 v	1.30 v	1.30 v	24.54 v	49.64 v	4.67 v
ESTUDIOFINAL57	(1)	0.00	0.17 v	0.17 v	1.85 v	1.85 v	24.56 v	50.06 v	5.66 v
ESTUDIOFINAL58	(1)	0.00	0.18 v	0.19 v	0.87 v	0.86 v	24.38 v	50.02 v	4.68 v
ESTUDIOFINAL59	(1)	0.00	0.16 v	0.17 v	1.15 v	1.16 v	24.27 v	52.95 v	5.20 v
Average		0.00	0.18	0.19	1.74	1.74	23.98	47.50	5.03
	(v/ /*)		(59/0/0)	(59/0/0)	(59/0/0)	(59/0/0)	(59/0/0)	(59/0/0)	(59/0/0)

Figura 4.9 - Tiempos de computación para los desvíos de la demanda con ventana móvil de 100 días

Como se comprueba, la media de tiempo que utiliza dicho árbol de decisión para generar un modelo de una muestra de 2400 datos, es de 0.18 segundos, lo que multiplicado por 59, hace un total de 10.62 segundos. En cambio la red neuronal con la configuración de parámetros elegida, tarda 47.50 segundos en generar un modelo, lo que hace un total de casi 47 minutos para realizar las 59 simulaciones.

A continuación se muestran (*figura 4.10*) los resultados de las simulaciones para los desvíos eólicos.

Dataset	(1)	rules.Zer	(2) trees.	(3) trees.	(4) meta.A	(5) meta.A	(6) funct	(7) funct	(8) meta.A
ESTUDIOFINAL01	(1)	12.50	87.50 v	87.50 v	12.50	12.50	87.50 v	79.17 v	83.33 v
ESTUDIOFINAL02	(1)	16.67	83.33 v	83.33 v	83.33 v	29.17 v	83.33 v	75.00 v	83.33 v
ESTUDIOFINAL03	(1)	16.67	62.50 v	62.50 v	37.50 v	41.67 v	83.33 v	70.83 v	50.00 v
ESTUDIOFINAL04	(1)	0.00	70.83 v	75.00 v	87.50 v	87.50 v	79.17 v	79.17 v	83.33 v
ESTUDIOFINAL05	(1)	33.33	66.67 v	50.00 v	33.33	33.33	58.33 v	66.67 v	87.50 v
ESTUDIOFINAL06	(1)	100.00	100.00	100.00	70.83 *	70.83 *	41.67 *	83.33 *	100.00
ESTUDIOFINAL07	(1)	87.50	87.50	87.50	70.83 *	70.83 *	87.50	87.50	87.50
ESTUDIOFINAL08	(1)	100.00	12.50 *	12.50 *	100.00	100.00	4.17 *	91.67 *	91.67 *
ESTUDIOFINAL09	(1)	70.83	58.33 *	58.33 *	70.83	70.83	29.17 *	70.83	70.83
ESTUDIOFINAL10	(1)	66.67	50.00 *	50.00 *	66.67	66.67	45.83 *	70.83 v	58.33 *
ESTUDIOFINAL11	(1)	66.67	70.83 v	70.83 v	66.67	66.67	54.17 *	33.33 *	50.00 *
⋮		⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
ESTUDIOFINAL47	(1)	91.67	62.50 *	33.33 *	91.67	91.67	45.83 *	66.67 *	8.33 *
ESTUDIOFINAL48	(1)	58.33	54.17 *	58.33	62.50 v	62.50 v	54.17 *	62.50 v	58.33
ESTUDIOFINAL49	(1)	16.67	41.67 v	50.00 v	16.67	33.33 v	37.50 v	41.67 v	50.00 v
ESTUDIOFINAL50	(1)	79.17	58.33 *	50.00 *	62.50 *	41.67 *	54.17 *	25.00 *	54.17 *
ESTUDIOFINAL51	(1)	20.83	29.17 v	29.17 v	8.33 *	8.33 *	37.50 v	12.50 *	20.83
ESTUDIOFINAL52	(1)	66.67	66.67	54.17 *	66.67	66.67	62.50 *	45.83 *	66.67
ESTUDIOFINAL53	(1)	45.83	45.83	54.17 v	45.83	45.83	45.83	66.67 v	45.83
ESTUDIOFINAL54	(1)	91.67	83.33 *	83.33 *	87.50 *	87.50 *	83.33 *	66.67 *	91.67
ESTUDIOFINAL55	(1)	100.00	79.17 *	79.17 *	100.00	75.00 *	75.00 *	58.33 *	100.00
ESTUDIOFINAL56	(1)	41.67	41.67	41.67	37.50 *	37.50 *	33.33 *	41.67	41.67
ESTUDIOFINAL57	(1)	29.17	58.33 v	45.83 v	33.33 v	33.33 v	70.83 v	58.33 v	54.17 v
ESTUDIOFINAL58	(1)	33.33	33.33	33.33	33.33	33.33	33.33	33.33	58.33 v
ESTUDIOFINAL59	(1)	75.00	58.33 *	58.33 *	75.00	50.00 *	54.17 *	41.67 *	41.67 *
Average		59.68	61.65	59.53	58.90	55.44	58.12	58.97	61.51
	(v/ /*)		(23/10/26)	(25/7/27)	(17/22/20)	(17/18/24)	(21/4/34)	(26/5/28)	(17/20/22)

Figura 4.10 - Resultados para los desvíos eólicos con ventana móvil de 100 días

Para el caso eólico, como ya se vio en el primero de los estudios utilizando las parejas de años, los resultados no son nada positivos. Exceptuando un par de ocasiones donde se supera ligeramente al clasificador *ZeroR*, el resto de configuraciones no generan ninguna mejoría sobre lo que ya aporta este método.

Por último, se muestran también (*figura 4.11*) los resultados para los desvíos de la generación fotovoltaica.

Dataset	(1) rules.Ze	(2) trees	(3) trees.	(4) meta.	(5) meta.	(6) funct	(7) funct	(8) meta.
ESTUDIOFINAL01	(1) 87.50	66.67 *	50.00 *	87.50	87.50	20.83 *	12.50 *	83.33 *
ESTUDIOFINAL02	(1) 79.17	75.00 *	66.67 *	79.17	79.17	79.17	54.17 *	45.83 *
ESTUDIOFINAL03	(1) 37.50	75.00 v	83.33 v	75.00 v	75.00 v	62.50 v	58.33 v	70.83 v
ESTUDIOFINAL04	(1) 37.50	75.00 v	70.83 v	75.00 v	79.17 v	75.00 v	58.33 v	70.83 v
ESTUDIOFINAL05	(1) 29.17	91.67 v	91.67 v	75.00 v	75.00 v	62.50 v	79.17 v	75.00 v
ESTUDIOFINAL06	(1) 66.67	66.67	70.83 v	70.83 v	75.00 v	75.00 v	58.33 *	75.00 v
ESTUDIOFINAL07	(1) 37.50	87.50 v	87.50 v	50.00 v	50.00 v	83.33 v	91.67 v	91.67 v
ESTUDIOFINAL08	(1) 70.83	66.67 *	66.67 *	83.33 v	87.50 v	70.83	45.83 *	66.67 *
ESTUDIOFINAL09	(1) 54.17	87.50 v	91.67 v	79.17 v	79.17 v	54.17	66.67 v	70.83 v
ESTUDIOFINAL10	(1) 79.17	41.67 *	41.67 *	79.17	79.17	45.83 *	50.00 *	54.17 *
ESTUDIOFINAL11	(1) 70.83	58.33 *	62.50 *	70.83	70.83	62.50 *	58.33 *	70.83
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
ESTUDIOFINAL47	(1) 66.67	66.67	70.83 v	66.67	66.67	70.83 v	75.00 v	66.67
ESTUDIOFINAL48	(1) 62.50	83.33 v	70.83 v	83.33 v	83.33 v	83.33 v	66.67 v	66.67 v
ESTUDIOFINAL49	(1) 79.17	79.17	83.33 v	70.83 *	70.83 *	75.00 *	87.50 v	75.00 *
ESTUDIOFINAL50	(1) 83.33	62.50 *	54.17 *	87.50 v	87.50 v	79.17 *	91.67 v	79.17 *
ESTUDIOFINAL51	(1) 75.00	91.67 v	87.50 v	83.33 v	83.33 v	87.50 v	91.67 v	83.33 v
ESTUDIOFINAL52	(1) 79.17	66.67 *	70.83 *	79.17	79.17	79.17	83.33 v	75.00 *
ESTUDIOFINAL53	(1) 83.33	79.17 *	83.33	75.00 *	75.00 *	87.50 v	83.33	75.00 *
ESTUDIOFINAL54	(1) 83.33	79.17 *	79.17 *	66.67 *	66.67 *	70.83 *	83.33	79.17 *
ESTUDIOFINAL55	(1) 75.00	70.83 *	79.17 v	75.00	70.83 *	87.50 v	70.83 *	75.00
ESTUDIOFINAL56	(1) 75.00	83.33 v	87.50 v	91.67 v	91.67 v	87.50 v	87.50 v	75.00
ESTUDIOFINAL57	(1) 62.50	58.33 *	58.33 *	79.17 v	79.17 v	62.50	58.33 *	70.83 v
ESTUDIOFINAL58	(1) 12.50	62.50 v	45.83 v	62.50 v	58.33 v	41.67 v	12.50	25.00 v
ESTUDIOFINAL59	(1) 58.33	79.17 v	75.00 v	75.00 v	75.00 v	79.17 v	66.67 v	79.17 v
Average	64.55	72.53	70.41	69.99	69.63	70.90	69.28	70.13
(v/ /*)		(31/5/23)	(33/3/23	(29/16/14	(31/13/15)	(35/7/17)	(32/10/17)	(31/9/19)

Figura 4.11 - Resultados para los desvíos fotovoltaicos con ventana móvil de 100 días

Como se podía prever, se dan resultados positivos y que se asemejan a los obtenidos para los desvíos del sistema y de la demanda. Aun así, se debería hacer un análisis más exhaustivos si se quiere comprobar si verdaderamente estos métodos mejoran en alguna medida, las estrategias desarrolladas en el anterior trabajo, ya que los desvíos de la fotovoltaica se sucedían de manera tan repetitiva que era fácil predecir las curvas con ayuda de la simple observación y quizás se podía esperar que con métodos más sofisticados mínimamente se igualasen los resultados.

5. CONCLUSIONES Y LÍNEAS FUTURAS

Una vez llegados a este punto, es importante hacer un repaso de las ideas generales del estudio y recopilar las que han podido ser las conclusiones más importantes.

En primer lugar, se vuelve a hacer hincapié en la idea de que este trabajo nace como continuación de un estudio anterior con una serie de conclusiones, las cuales se han utilizado como base para muchas de las ideas e hipótesis aquí planteadas. Tanto en este estudio, como en el anterior, todos los resultados se han obtenido partiendo de datos reales, los cuales han sido tratados con especial atención. Esta última aclaración resulta realmente importante cuando se tratan cantidades tan grandes de datos. Si la información de las distintas variables tiene un origen poco fiable, o se cometen errores que hacen emparejar datos que realmente no coinciden en el tiempo, los resultados no serían reales y las conclusiones obtenidas, falsas.

Tras una breve introducción al estudio, se ha hecho una descripción y un recorrido por el programa utilizado. Posteriormente se han establecido diferentes hipótesis y casos de estudio y se han obtenido una serie de resultados, más o menos positivos dependiendo del caso. Con todo ello, se ha podido comprobar del gran potencial que tiene WEKA. Aunque solo se hayan explicado las herramientas principales y dos de sus métodos de predicción, el programa cuenta con más herramientas, más algoritmos de predicción y otros puntos de vista desde los que aplicar los métodos ya utilizados.

Ya se vio en el anterior trabajo, que a nivel económico, resultaba interesante a nivel privado de cada sujeto de mercado, el poder prever el sentido del desvío del sistema, ya que, tras una serie de casos prácticos con datos reales, se veía que el nivel de pérdidas era significativo. Concretamente, se exponía uno caso hipotético de una central eólica de 100 MW. Para esta hipotética central, se suponía una generación y unos desvíos anuales iguales a los de la totalidad de la generación eólica española pero guardando proporcionalidad con sus 100 MW instalados. Dentro de estas condiciones y con datos reales sobre los desvíos, esta central hubiera tenido unas pérdidas de 91.872,00 € para 2016. Simplemente haciendo una observación de los resultados de ese año, se proponían una serie de factores por los que multiplicar las ofertas realizadas por la central para tratar de disminuir los desvíos. Una vez conocidos los resultados del año y aplicando los factores a posteriori, se vio que se podría haber obtenido un 7.25 % de ahorro, pero esto se obtuvo una vez se conocían los datos de lo sucedido. Sería posible realizar un estudio económico de lo que podría haberse ahorrado en distintas épocas utilizando la metodología de predicción del actual trabajo, pero ello requeriría de un estudio en profundidad, nueva descarga y tratamiento de datos y realizar nuevas predicciones lo cual no es objeto de este proyecto, quedando esta idea para desarrollos futuros. Aun así, es posible de manera sencilla aportar ciertos datos que dan una idea de la utilidad de los resultados obtenidos.

Para el caso de la central eólica, se ha dicho que las pérdidas en penalizaciones ascendieron a 91.872,00 €. Se conoce que esta central cometió 6044 desvíos a favor de los desvíos del sistema, es decir, perjudicándolo. Estos desvíos son los que trajeron aparejados las pérdidas citadas. Utilizando el método de los factores que conseguía una reducción de 7.25%, se disminuían los desvíos cometidos en contra del sistema a 5476 desvíos. En este estudio, se ha desarrollado una

metodología que obtiene una media de aciertos en las predicciones por encima del 70 %, lo cual en un primer momento, reduce los desvíos en contra del sistema a un máximo de 2628 desvíos. Esto corresponde a haber podido reducir los desvíos cometidos en un 56,5 %. No se conocen cifras económicas reales porque entran en juego el precio de los desvíos para cada hora y los distintos casos que se puedan dar, por lo tanto no resulta sencillo ni fiable realizar una ponderación, debería hacerse un estudio más completo. Aun así, a simple vista la mejoría es considerable. Además hay que tener en cuenta que esto es una primera aproximación y que se han dejado vías abiertas para un estudio más exhaustivo, donde se haga un seguimiento día a día, se obtengan las distintas matrices de confusión y se apliquen factores de penalización a las horas donde se conozcan los fallos que se están cometiendo.

Una idea que resulta también importante, es la de que el sistema eléctrico está en constante evolución. Continuamente surgen nuevos decretos y regulaciones. Hace 20 años el mercado aún estaba por liberalizar, hace pocos años que las renovables han dejado de ofertar a cero en los mercados y ya se está viendo la derogación del impuesto al sol. Esta constante evolución hace que las reglas y las variables de importancia en la red cambien cada año, por lo que lo primordial, será conocer que parámetros son importantes para determinar el comportamiento de la variable que se esté estudiando y contar con datos actualizados de ellos.

6. BIBLIOGRAFÍA

La totalidad de la bibliografía consultada para el desarrollo del proyecto se encuentra en los procedimientos de Red Eléctrica de España que más adelante se enumeran y en el manual de usuario de WEKA. Todos los datos descargados relativos al sistema eléctrico español y a la información meteorológica han sido descargados de www.esios.ree.es y www.meteomanz.com respectivamente.

- WEKA Manual for versión 3-8-2
- P.O.1.5 Establecimiento de la Reserva para la Regulación Frecuencia-Potencia
- P.O.2.1 Previsión de la Demanda
- P.O.3.1 Programación de la Generación
- P.O.3.2 Restricciones Técnicas
- P.O.3.3 Gestión de Desvíos
- P.O.7.2 Regulación Secundaria
- P.O.7.3 Regulación Terciaria
- P.O.14.4 Derechos de Cobro y Obligaciones de Pago por los Servicios de Ajuste del Sistema